



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

PROYECTO FIN DE CARRERA

INGENIERÍA TÉCNICA INDUSTRIAL: ELECTRÓNICA INDUSTRIAL

**"DISEÑO DE UN SISTEMA DE CONTROL DE
DIRECCIÓN DE UN VEHÍCULO"**

Autor: **Daniel Adrados Espada**

Tutor: **Luis Moreno Llorente**

JUNIO 2012



AGRADECIMIENTOS

Sobre todo agradecer a mis padres de los que he podido recibir toda mi educación y que han estado apoyándome en todo momento. Mi madre ha sabido cuidar de mí siempre perfectamente y me ha enseñado los valores y principios que ahora poseo. Mi padre siempre ha sido un apoyo y una referencia en los estudios, ayudándome y enseñándome sus conocimientos y su forma de plantear los problemas; sin él, no habría podido llegar a adquirir las bases fundamentales de las matemáticas, así como la lógica para saber afrontar los problemas.

Agradecer a mi hermana que se que me aprecia y me quiere un montón y al resto de mi familia por como son y por el cariño y el amor que recibo siempre de ellos.

Destacar a mi abuelo Daniel, que para mí siempre fue un modelo como persona, la mejor y más bondadosa que he conocido nunca.

Agradecer a Kiara, Gloria y Nilda por darme los mejores años de mi vida junto a ellas, donde he encontrado el amor y la más absoluta felicidad.

No se me olvidan mis amigos, que se que siempre están ahí, en los malos y buenos momentos. Tampoco se me olvidan mis compañeros de fatigas durante estos años de carrera, sin ellos no habría sido posible llegar hasta el final.

Por su puesto agradecer a mi tutor por brindarme la oportunidad de poder realizar mi proyecto de fin de carrera con él.

Gracias a todos.

RESUMEN

El proyecto consta de dos partes diferenciadas, en la primera se desarrolla un sistema de control de dirección para una moto de Lego, basado en el seguimiento de una línea negra sobre una plantilla de color blanco. Para ello se compró un paquete de Lego Mindstorm NXT, con el que hemos construido la moto que se va a usar. Esta lleva instalada una cámara de video la cual capta las imágenes de la trayectoria de la línea negra que deberá seguir la moto y las envía al ordenador. El ordenador procesa las imágenes y envía por Bluetooth el valor del ángulo que debe girar la moto, esta variable es recibida por la CPU del Lego, que manda la señal de giro al motor paso a paso de la dirección. Este ciclo de control se va repitiendo en intervalos de décimas de segundo y de esta forma la moto se va guiando a través de la línea negra. El procesamiento de las imágenes se hace mediante el programa Matlab, que nos proporciona las herramientas necesarias para el tratamiento digital de imágenes. El envío de órdenes y la conexión del ordenador con la CPU de Lego se hacen posibles gracias al paquete de funciones, desarrollado por la Universidad de Aachen, que permite programar La CPU de Lego desde el Matlab.

En la segunda parte del proyecto se lleva a cabo el cálculo de la inclinación de la moto, generando una señal que podría ser procesada posteriormente por un sistema de control de estabilidad. Para ello se han instalado en la moto un par de punteros láser, uno a cada lado de la moto, que se toman como puntos de referencia para obtener la inclinación. Las proyecciones de estos punteros son captadas por la cámara de la moto y varían de posición en la imagen según la inclinación de la moto.

1. INDICE

AGRADECIMIENTOS.....	3
RESUMEN	5
1. INDICE.....	6
2. MOTIVACIÓN Y OBJETIVOS	9
3. ESTADO DEL ARTE	11
3.1. VISIÓN ARTIFICIAL	11
3.1.1. DEFINICIÓN DE VISIÓN ARTIFICIAL.....	11
3.1.1.2. DEFINICIONES DADAS POR LOS INVESTIGADORES A LO LARGO DE LA HISTORIA	14
3.1.2. VISIÓN HUMANA VS VISIÓN ARTIFICIAL	15
3.1.3. USOS.....	18
3.1.3.1. CONTROL DE PROCESOS.....	20
3.1.3.2. CONTROL DE CALIDAD.....	21
3.1.3.3. Aplicaciones no industriales.....	24
3.1.4. COMPONENTES DE UN SISTEMA DE VISIÓN ARTIFICIAL.....	25
3.1.4.1. EL SISTEMA DE ILUMINACIÓN	29
3.1.4.1.1. FUENTES DE LUZ.....	29
3.1.4.1.2. ESQUEMAS DE ILUMINACIÓN	33
3.1.4. 2. LA CÁMARA	35
3.1.4. 2.1. SENSORES DIGITALES DE IMAGEN.....	35
3.1.4. 2.2. OBJETIVOS	37
3.1.5. CÓMO FUNCIONA LA VISIÓN ARTIFICIAL	39
3.1.5.1. ADQUISICIÓN DE LA IMAGEN.....	40
3.1.5.2. PREPROCESADO	40

3.1.5.3.	SEGMENTACIÓN	41
3.1.5.4.	REPRESENTACIÓN Y DESCRIPCIÓN	41
3.1.5.5.	RECONOCIMIENTO E INTERPRETACIÓN	41
3.1.6.	JUSTIFICACIÓN DE LA INVERSIÓN EN VISIÓN ARTIFICIAL.....	42
3.1.6.1.	COSTES DE LOS MATERIALES.....	43
3.1.6.2.	COSTES DE LA MANO DE OBRA	43
3.1.6.3.	COSTES DE LA CALIDAD	43
3.2.	LA VISIÓN ARTIFICIAL EN LOS VEHÍCULOS	44
4.	PLANTEAMIENTO Y MEDIOS UTILIZADOS	48
4.1.	PLANTEAMIENTO	48
4.2.	MEDIOS UTILIZADOS	49
4.2.1.	HARDWARE	49
4.2.1.1.	LEGO MINDSTORM NXT	49
4.2.1.2.	Microsoft LifeCam VX-800.....	52
4.2.1.3.	PUNTEROS LASER	54
4.2.1.4.	MODELO DE PRUEBAS DE LA MOTO	55
4.2.1.5.	DISPOSITIVO BLUETOOTH	55
4.2.2.	SOFTWARE.....	56
4.2.2.1.	MATLAB.....	56
5.	IMPLEMENTACIÓN	59
5.1.	CONTROL DE DIRECCIÓN.....	59
5.1.1.	SISTEMA DE ADQUISICIÓN DE IMÁGENES	60
5.1.2.	CONFIGURACIÓN Y CONEXIÓN DEL "BRICK" DE LEGO	61
5.1.3.	BUCLE DE CONTROL	61
5.1.3.2.	CONTROL DE GIRO.....	70
5.2.	CALCULO DE LA INCLINACION DE LA MOTO	72
5.2.1	COLOCACIÓN DE LOS PUNTEROS LASER	73
5.2.2.	RECONOCIMIENTO DE LOS PUNTEROS LASER	75
6.	RESULTADOS Y CONCLUSIONES	84
7.	ANEXOS	85
7.1.	CODIGO	85
7.1.1.	CÓDIGO: control_dir.m	85
7.1.2.	CÓDIGO: calc_giro.m.....	87



7.1.3. CÓDIGO: calc_inclinacion.m.....	91
7.2. SECUENCIAS DE IMÁGENES CAPTURADAS EN EL PROCESO.....	93
7.3. VIDEO DEL SEGUIMIENTO DE LA LÍNEA NEGRA POR LA MOTO.....	105
8. BIBLIOGRAFIA.....	106

2. MOTIVACIÓN Y OBJETIVOS

En la universidad se desarrolla un gran trabajo en el ámbito de la robótica, con una gran variedad de proyectos e investigaciones, llevados a cabo por estudiantes en sus proyectos de fin de carrera o por estudiantes de postgrado que están haciendo un doctorado. De uno de esos proyectos teníamos a nuestra disposición una moto de radio control.

Éramos dos los compañeros que estábamos interesados en hacer un proyecto relacionado con la robótica, y nos gustaba mucho la idea de que nuestra base de trabajo fuese una moto de radio control. De aquí surgen dos ideas, para dos proyectos distintos. Mi proyecto consistiría en implementar un control de dirección autónomo, instalando una cámara y un microprocesador en la moto y mi compañero se encargaría de desarrollar un sistema de control de estabilidad. Para el control de estabilidad necesitamos calcular la inclinación de la moto en todo momento, para lo que él usó un giróscopo. Como complemento al sistema de control de estabilidad se nos ocurrió intentar obtener la inclinación mediante el uso de la cámara que yo iba a instalar en la moto.

El uso de la moto de radio control lo tuvimos que descartar posteriormente ya que la moto se encontraba en mal estado y no funcionaba correctamente; así que compramos el kit de robótica Lego NXT Mindstorm, el cual te permite la construcción de distintos tipos de robots, entre ellos una moto.

De estas ideas surgen los dos objetivos del proyecto.

- El primer objetivo es conseguir el control direccional de la moto de forma autónoma. Este control se ha llevado a cabo mediante el desarrollo de un

sistema de reconocimiento y seguimiento de una línea negra trazada en el suelo.

- El segundo objetivo es calcular la inclinación de la moto en tiempo real y generar una señal que se pueda procesar posteriormente en un sistema de control de estabilidad.

Para el sistema de control direccional, se procesarán las imágenes de la línea negra obtenidas por una cámara de video instalada en la moto. Para el cálculo de la inclinación nos ayudaremos de dos punteros laser instalados en la moto, de tal forma que la misma cámara capte la incidencia de los punteros en el suelo cuyo tratamiento posterior determinará la inclinación de la moto en todo momento.

El desarrollo de este proyecto se basa en el procesamiento de imágenes por ordenador, aplicado a la robótica se conoce como visión artificial, ya que se usa para dotar a una máquina o dispositivo de una capacidad de visión "humana". Este campo tiene multitud de aplicaciones en la industria y cada vez está llegando más a la tecnología de uso personal. Lo podemos encontrar en el control de calidad de una empresa alimentaria, comprobando mediante unos patrones que todos sus productos tengan la forma y dimensión adecuada; también lo podemos utilizar para contar las personas que acuden a una manifestación, o como se está viendo últimamente aplicado a la industria automovilística, con coches que mediante unas cámaras colocadas en la parte frontal, son capaces de proporcionarnos la lectura de las señales de tráfico y adaptar la velocidad del coche automáticamente, o impedir que nos cambiemos de carril involuntariamente. Sin olvidarnos de los nuevos *smartphones*, que desarrollan multitud de aplicaciones usando la cámara incluida en el teléfono para el procesamiento de imágenes; un ejemplo típico es la activación del disparador de la cámara de fotos del móvil, cuando esta detecta una sonrisa en la persona que se le está haciendo la foto.

El procesamiento de imágenes por ordenador es un campo que siempre me ha atraído mucho y que en la carrera no se trata en profundidad. Por ello me adentré en este proyecto, aún no teniendo mucha idea del tema, para así poder incrementar mis conocimientos y habilidades en el desarrollo de programas que utilicen esta técnica.

3. ESTADO DEL ARTE

3.1. VISIÓN ARTIFICIAL

Uno de los sentidos más importantes de los seres humanos es la visión. Ésta es empleada para obtener la información visual del entorno físico. Según Aristóteles, “Visión es saber que hay y donde mediante la vista”. De hecho, se calcula que más de 70% de las tareas del cerebro son empleadas en el análisis de la información visual. El refrán popular de “Una imagen vale más que mil palabras” tiene mucho que ver con los aspectos cognitivos de la especie humana. Casi todas las disciplinas científicas emplean utillajes (conjunto de utensilios y materiales necesarios para llevar a cabo una tarea) gráficos para transmitir conocimiento. Por ejemplo, en Ingeniería Electrónica se emplean esquemas de circuitos, a modo gráfico, para describirlos. Se podría hacerlo mediante texto, pero para la especie humana resulta mucho más eficiente procesar imágenes que procesar texto. La visión humana es el sentido más desarrollado y el que menos se conoce debido a su gran complejidad. Es una actividad inconsciente y difícil de saber cómo se produce. De hecho, hoy en día, se carece de una teoría que explique cómo los humanos perciben el exterior a través de la vista.

3.1.1. DEFINICIÓN DE VISIÓN ARTIFICIAL

La visión artificial es una rama de la inteligencia artificial (ver figura 3.1) que tiene por objetivo modelar matemáticamente los procesos de percepción visual en los seres vivos y generar programas que permitan simular estas capacidades visuales por computadora. La visión artificial permite la detección automática de la estructura y propiedades de un posible mundo dinámico en 3 dimensiones a partir una o varias imágenes bidimensionales del mundo. Las imágenes pueden ser monocromáticas o a

color; pueden ser capturadas por una o varias cámaras, y cada cámara puede ser estacionaria o móvil. La estructura y propiedades del mundo tridimensional que se intentan deducir en la visión artificial incluyen no sólo propiedades geométricas (tamaños, formas, localización de objetos, etc.), sino también propiedades del material (sus colores, sus texturas, la composición, etc.) y la luminosidad u oscuridad de las superficies.

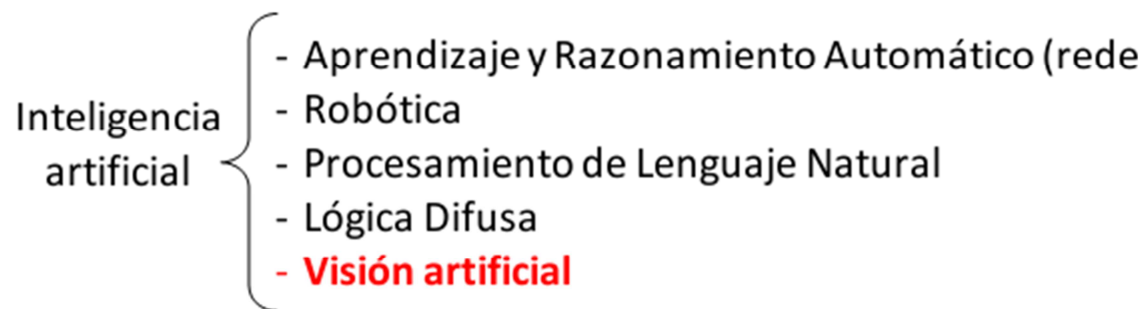


Figura 3.1: Ramas de la inteligencia artificial

Los sistemas de percepción computacional, como también se conoce a la visión artificial, van más allá de medir o detectar, estos sistemas perciben, es decir descifran o reconocen el mensaje sensorial (figura 3.2). La información visual es una proyección bidimensional de objetos tridimensionales y, por tanto, la imagen que capta el ojo humano o una cámara digital tiene infinitas interpretaciones posibles. La percepción es un proceso que se distribuye a lo largo del espacio y del tiempo.

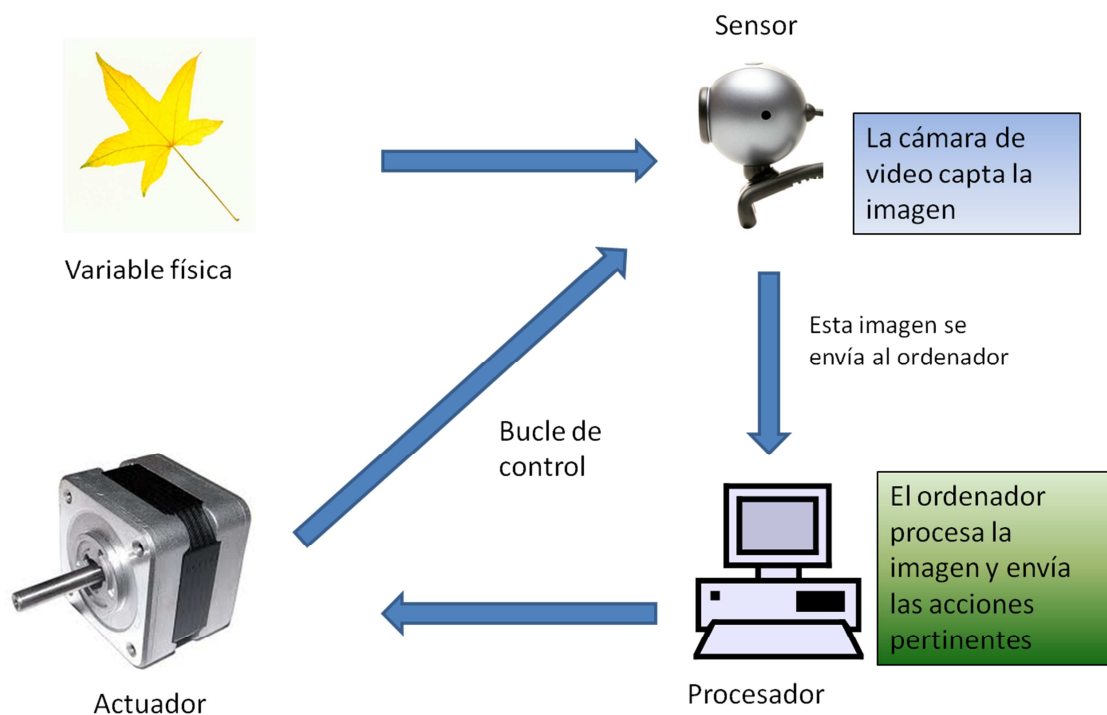


Figura 3.2: Bucle de control

La Visión Artificial, pretende capturar la información visual del entorno físico para extraer características relevantes visuales, utilizando procedimientos automáticos. Según Marr, "Visión es un proceso que a partir de imágenes del mundo exterior produce una descripción útil para el observador y no tiene información irrelevante".

La visión artificial aplicada a la industrial abarca la informática, la óptica, la ingeniería mecánica y la automatización industrial. A diferencia de la visión artificial académica, que se centra principalmente en máquinas basadas en el procesamiento de imágenes, las aplicaciones de visión artificial industrial integran sistemas de captura de imágenes digitales, dispositivos de entrada/salida y redes de ordenador para el control de equipos destinados a la fabricación tales como brazos robóticos. Los sistemas de visión artificial se destinan a realizar inspecciones visuales que requieren alta velocidad, gran aumento, funcionamiento las 24 horas del día o la repetitividad de las medidas.

El objetivo de un sistema de inspección por visión artificial suele ser comprobar la conformidad de una pieza con ciertos requisitos, tales como las dimensiones, números de serie, la presencia de componentes, etc.

3.1.1.2. DEFINICIONES DADAS POR LOS INVESTIGADORES A LO LARGO DE LA HISTORIA

Ballard&Brown (1982): La Visión Artificial es la construcción de descripciones de objetos físicos, llenas de significado y explícitas, a partir de imágenes, pg. xiii. Incluyen el procesamiento de imágenes (no pensamos que sea acertado en la actualidad).

Horn (1986) en “Robot Vision” página 4: la entrada a un sistema de visión por máquina es una imagen o varias imágenes mientras que la salida es una descripción que debe cumplir dos criterios: estar relacionada con la imagen observada y contener toda la información que se necesita para realizar una tarea determinada. El procesamiento de imágenes es una parte de esta tarea.

Nalwa (1993): La Visión Artificial describe la deducción automática de las estructuras y propiedades de un mundo tridimensional, posiblemente dinámico, a partir de una o varias imágenes bidimensionales de él.

El autor excluye de su libro el reconocimiento de objetos basado en modelos, ya que, según él, este reconocimiento es un proceso cognitivo y no de percepción.

Trucco&Verri (1998): La búsqueda de propiedades del mundo 3-D a partir de una o más imágenes digitales. Es importante notar que en este texto sí se incluye el reconocimiento de objetos.

Hoffman (1998): Tal vez estemos en una época en la que el reconocimiento de objetos sea parte de la VA, pero no sería extraño que en el futuro asistiéramos a una separación de esta materia de la VA como ya la hubo del procesamiento de imágenes y el análisis de imágenes. El mundo de la VA es fundamentalmente un mundo 3-D

3.1.2. VISIÓN HUMANA VS VISIÓN ARTIFICIAL

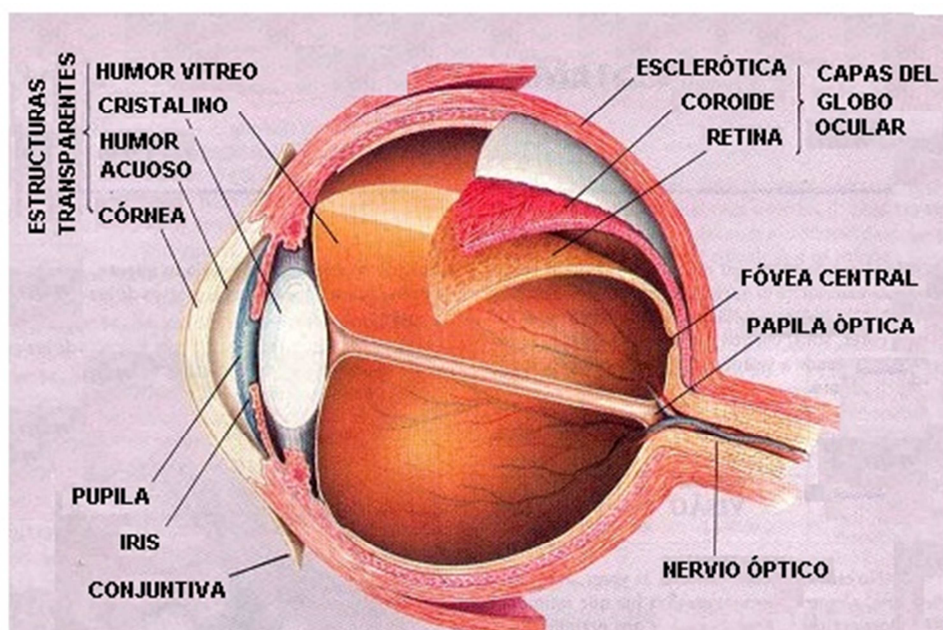


Figura 3.3: Partes del ojo humano

El ojo tiene una forma, aproximadamente, esférica de unos 2.5 cm de diámetro. Está formado por una óptica y una zona sensorial (ver figura 3.3). La óptica está constituida por la córnea, el iris o pupila y el cristalino. La córnea es un material transparente y funciona como lente fija. La pupila regula la cantidad de luz que entra en el interior y el cristalino hace las veces de lente variable, permitiendo el enfoque dependiendo de la distancia de los objetos. Los músculos que sujetan al cristalino provocan su deformación, cuya consecuencia es el enfoque del campo visual. La misión de la óptica del ojo es conducir la radiación electromagnética, del espectro visible, hacia la retina. La retina se encuentra en la parte posterior del glóbulo ocular y es donde se ubican las células fotosensibles: los conos y los bastones.

El hombre ha imitado muchas veces, en la construcción de sus artefactos, a la Naturaleza. En este caso también se cumple. Las cámaras de vídeo con sus ópticas hacen las veces del globo ocular, mientras el computador realizará las tareas de procesamiento, emulando el comportamiento del cerebro. Cuando se establecieron en la década de los 50, los objetivos de la Inteligencia Artificial, se suponía que con la

llegada del siglo XXI habría máquinas que serían capaces de describir, con información de alto nivel, las escenas capturadas. Con el paso del tiempo se vio que aquel anhelo se iba desvaneciendo. Hoy en día, todavía no hay una teoría de la visión. No se conoce los mecanismos que el cerebro utiliza para obtener la información de la percepción. El cerebro es capaz, de manera inconsciente, de determinar la distancia a los objetos, de reconocerlos en diferentes posiciones, aunque se encuentren rotados y con información parcialmente oculta. En definitiva, el cerebro presenta una sofisticación en la percepción que ni ahora ni en mucho tiempo habrá posibilidad de implementar artificialmente.

Lo que sí hace la Visión Artificial es construir nuevos y más sofisticados algoritmos que sean capaces de obtener información de bajo nivel visual. Y aunque todavía se esté años luz de la percepción visual de los seres vivos, la Visión Artificial es muy eficaz en tareas visuales repetitivas y alienantes para el hombre. Por ejemplo, en el campo de la inspección de productos en la industria o en contar células en una imagen de microscopía o en determinar la trayectoria de un vehículo en una autopista, etc.

El sentido de la vista nos proporciona el 75% de la información que recibimos los seres humanos en nuestro cerebro. Si las máquinas dispusieran de esta fuente de información con un sistema de percepción artificial, se les proporcionaría una mejora significativa en su funcionamiento y operación. El sistema de visión humano es maravilloso, sin embargo, no es infalible, en ciertas circunstancias puede ocasionar ilusiones que nos hace ver cosas que no son. Además, puede ser ambiguo pues nos da más de una interpretación posible y puede ser inconsistente informándonos de objetos y figuras imposibles en mundo real. A pesar del gran avance científico y tecnológico en este campo, los sistemas de visión artificial aun están muy lejos de alcanzar a los sistemas de visión humana, aunque no es necesario que se emule en todos los aspectos a este. Sin embargo, en los procesos tecnológicos y dentro de estos los procesos de producción, los sistemas de visión artificial pueden hacer cosas de manera mucho más efectivas y adecuadas que la visión humana, tal es el caso de los siguientes aspectos:

1. Dentro del espectro electromagnético (ver figura 3.4) la visión humana solamente capta un pequeño rango de frecuencias y amplitudes, conocido como “rango de luz visible”, los sistemas de visión artificial pueden captar todo el espectro, es decir, además del rango de luz visible puede captar ondas de radio, de televisión, microondas, infrarrojos, ultravioletas, rayos X, rayos gamma y rayos cósmicos.
2. El ser humano puede discernir entre 10 ó 20 niveles de gris, los sistemas de visión artificial tienen una definición muy superior.
3. La velocidad de respuesta de la visión humana es de 0,06 segundos, mientras que en las cámaras de estado sólido es de 0,00001 segundos y este tiempo se va reduciendo según se mejora la electrónica de estos sistemas.
4. La visión humana se cansa y se ve afectada por las emociones, en los sistemas de visión artificial no.
5. El ser humano puede discernir entre 10 ó 20 niveles de gris, los sistemas de visión artificial tienen una definición muy superior.
6. La visión humana tiene muy poca precisión apoyándose en instrumentos de medida, los sistemas de visión artificial tienen gran precisión en la medición, dependiendo solamente de la resolución espacial de los componentes del sistema.
7. La visión humana es poco consistente por la fatiga y las distracciones, en cambio la visión artificial mantiene su nivel de rendimiento constante a lo largo de su vida útil.

8. Los sistemas de visión artificial pueden trabajar en entornos muy peligrosos, con riesgos radioactivos, químicos, biológicos, ruido, contaminación, temperaturas muy altas y muy bajas.

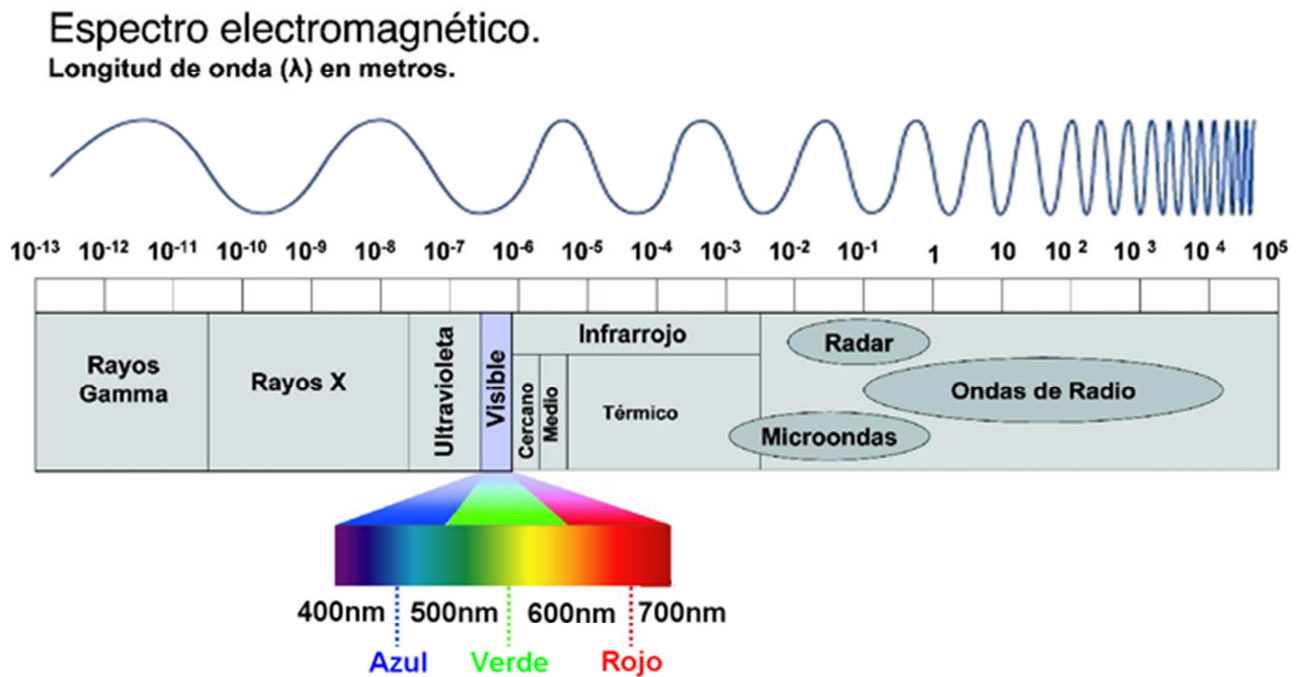


Figura 3.4: Espectro electromagnético

Las principales ventajas de la visión humana respecto a la artificial son:

- Mejor reconocimiento de objetos.
- Mejor adaptación a situaciones imprevistas.
- Mejor en tareas de alto nivel de proceso.
- Mayor resolución, 576Mpíxeles frente a los 50Mpíxeles de una cámara.

3.1.3. USOS

Con el incremento de potencia de los microprocesadores, las aplicaciones que pueden resolverse con éxito utilizando sistemas de visión están creciendo rápidamente. Una

aplicación puede exigir una o más funciones de procesamiento de imágenes, que cuando se combinan crean una solución.

La utilización de la visión artificial en la industria abarca un amplio espectro de aplicaciones que, de manera simplificada, pueden agruparse en cinco áreas principales:

- Control de procesos.
- Control de calidad.
- Aplicaciones no industriales.

Prácticamente cualquier sector industrial puede beneficiarse de esta tecnología para acelerar sus procesos, para reducir sus costes de producción o para mejorar la calidad de sus productos finales.

La visión artificial también es usada en aplicaciones no industriales como, por ejemplo, para el control del tráfico, guiado de robots, etc.

Los objetivos típicos de la visión artificial incluyen:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes (por ejemplo, caras humanas).
- La evaluación de los resultados.
- Registro de diferentes imágenes de una misma escena u objeto, esto es, hacer concordar un mismo objeto en diversas imágenes.
- Seguimiento de un objeto en una secuencia de imágenes.
- Mapeo de una escena para generar un modelo tridimensional de la escena; tal modelo podría ser usado por un robot para navegar por la escena.
- Estimación de las posturas tridimensionales de humanos.
- Búsqueda de imágenes digitales por su contenido.
- Estos objetivos se consiguen por medio de reconocimiento de patrones, aprendizaje

estadístico, geometría de proyección, procesamiento de imágenes, teoría de gráficos y otros campos. La visión artificial cognitiva está muy relacionada con la psicología cognitiva y la computación biológica.

3.1.3.1. CONTROL DE PROCESOS

Forma o apariencia. Control de la conformidad.

Los sistemas actuales, por lo general, comienzan con una operación de medición bidimensional para establecer el desplazamiento necesario para que el objeto se encuentre en una posición ideal.

Posteriormente, se llevan a cabo las operaciones de comparación con un objeto patrón o la caracterización geométrica de su

forma. De esta manera, se puede decidir si el objeto analizado cumple o no con los requisitos preestablecidos. La moneda de la figura 3.5 sería un claro ejemplo de defecto de fabricación que se podría detectar mediante visión artificial.



Figura 3.5: Moneda con un defecto de fabricación

Control por colores.

La inspección por color se utiliza ampliamente en la industria farmacéutica para confirmar que la tableta o píldora de color se ha colocado en el embalaje correcto.

Reconocimiento de piezas o componentes del producto.

El reconocimiento de piezas o componentes puede ser usado para identificar qué partes deben ser sometidas a que proceso. Por ejemplo, un sistema de visión puede discernir entre las diferentes piezas de un elemento e instruir a un robot para el correcto montaje del elemento.



Figura 3.6: Escáner láser leyendo un código de barras

Reconocimiento de caracteres.

Los sistemas de visión también pueden ser usados para leer códigos de barras con un mayor grado de confianza que un escáner láser (ver figura 3.6). Incluso si una gran parte del código de barras está oculto por suciedad, un sistema de visión, por lo general, puede leerlo, mientras que un

lector de códigos de barras convencional no.

Además de códigos de barras, los sistemas de visión también se usan para leer códigos matriciales de dos dimensiones, llamados códigos QR (ver figura 3.7), donde los códigos pueden contener mucha más información que un código de barras unidimensional.



Figura 3.7: Código QR

3.1.3.2. CONTROL DE CALIDAD

Hoy en día, en muchos procesos de fabricación, los límites de detección de defectos han superado la percepción del ojo humano; por esta razón las empresas del mundo moderno han visto la necesidad de crear un sistema que permita controlar en forma precisa y acertada la calidad de sus productos. Todo proceso industrial es evaluado por la calidad de su producto final, esto hace de la etapa de control de calidad una fase

crucial del proceso. Los mecanismos utilizados para establecer la calidad de un producto varían dependiendo de los parámetros que tengan relevancia en el mismo. Cuando el parámetro relevante es la geometría o forma del objeto fabricado se suele dejar a la vista del operario que lleve a cabo tal función tanto de inspección como de verificación para el control de calidad, sin embargo pueden existir errores en la geometría de un objeto que escapen de la vista de un operario y que luego impidan el buen funcionamiento de dicho objeto. En un caso como éste, surge como una buena alternativa el utilizar un sistema de visión artificial capaz de detectar aquellos errores que un operario pudiera pasar por alto. Los beneficios que se obtienen con esta tecnología hablan por sí solos:

- Mejora de la calidad.
- Menor nivel de rechazo del producto acabado.
- Reducción del costo de mano de obra.
- Agrupación en lotes según diferentes niveles de calidad.
- Análisis detallado de los fallos.
- Fácil adaptación a modificaciones en el producto o ritmo de fabricación.

Detección de defectos. Elementos discretos.

La detección de defectos puede ser considerada como un caso especial de control de la conformidad, en el que los objetos no ofrecen formas específicas caracterizables. Como ejemplos, citaremos la inspección de baldosas cerámicas en la etapa previa al esmaltado para detectar defectos en la superficie como rugosidades, o el uso de infrarrojos para la inspección de defectos de botellas de vidrio.

Detección de defectos. Tejidos o laminados.

La característica de muchos sistemas de este tipo es que se debe examinar grandes áreas a gran velocidad, en busca de defectos relativamente pequeños. El material producido en bobinas incluye la chapa de acero, papel, películas fotográficas, vidrio,

hoja de plástico y los textiles. Algunos de estos sistemas utilizan el escaneado láser en lugar de cámaras lineales para formar la imagen.

Medidas unidimensionales o bidimensionales.

Las mediciones bidimensionales se pueden realizar mediante la superposición de calibres ópticos sobre la imagen del producto. La medición por este método es muy rápida porque no hay movimiento mecánico, y el posicionamiento exacto del objeto a medir no es crítico (el sistema puede determinar el desplazamiento X e Y del objeto y el grado de rotación).

Algunos sistemas ofrecen interpolación sub píxel que, en condiciones adecuadas, hace que el sistema sea capaz de medir y, en particular, detectar cambios o diferencias en la medición, de milésimas en lugar de las centésimas que ofrecen otros métodos de metrología (ver ejemplo 1). El número de píxeles de la imagen se puede aumentar mediante el uso de cámaras de muy alta resolución y obtener así mediciones aun más precisas.

Ejemplo 1: Aplicación de interpolación subpixel:



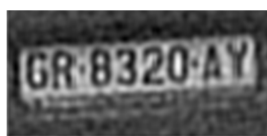
Imagen interpolada de 128x64 a partir de una imagen 32x16



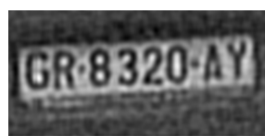
Reconstrucción con una imagen de baja resolución



Reconstrucción con dos imágenes de baja resolución



Reconstrucción con 4 imágenes de baja resolución



Reconstrucción con 8 imágenes de baja resolución



Reconstrucción con 16 imágenes de baja resolución

3.1.3.3. Aplicaciones no industriales

Guiado predeterminado.

El guiado predeterminado se caracteriza por una situación en la que una cámara acimutal (que le permite moverse en horizontal y vertical) toma una instantánea de la escena y el sistema de visión dirige un robot para que recoja o deje un objeto en una posición concreta. A continuación, el robot obra a ciegas. Una aplicación típica incluye tareas de paletización (colocar mercancía sobre un palé) de objetos pesados, tales como bloques de motor y cigüeñales, pero también puede incluir trabajos mucho más ligeros tales como el embalaje de chocolates.

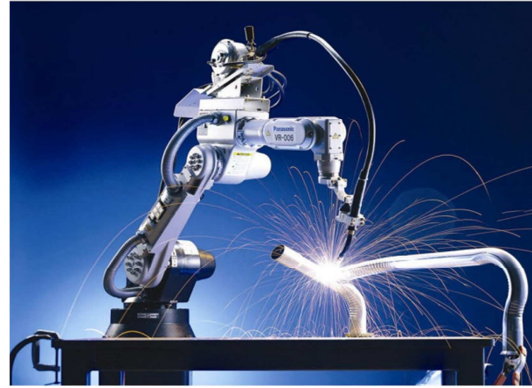


Figura 3.8: Robot soldando mediante arco eléctrico

Guiado continuo.

El guiado continuo implica una cámara montada sobre el brazo o la mano del robot y el camino del robot está continuamente corregido por el sistema de visión. Una aplicación común es el guiado de la soldadura por arco (ver ilustración 6), pero la técnica puede utilizarse también para controlar el camino en la aplicación de selladores o pegamentos por robots.

3.1.4. COMPONENTES DE UN SISTEMA DE VISIÓN ARTIFICIAL



Figura 3.9: Sistema de iluminación

Un moderno sistema de visión artificial consta de:

Un sistema de iluminación.

Es un aspecto de vital importancia ya que debe de proporcionar unas condiciones de iluminación uniformes e independientes del entorno, facilitando además, si es posible, la extracción de los rasgos de interés para una determinada aplicación.

Las variaciones en la iluminación son percibidas por el sistema como variaciones en los objetos. Es necesario conseguir una iluminación estable que resalte (incremente el contraste) de los elementos a detectar y evite sombras y reflejos. Intentar resolver tareas de alta precisión mediante una imagen de baja calidad es un problema que se suele intentar compensar con complejos algoritmos que ralentizan el sistema y no acaban resolviendo del todo el problema de base.

Una buena iluminación es especialmente importante para la toma de imágenes de los productos en una línea rápida de producción, aunque algunas aplicaciones pueden utilizar la luz ambiente. En la figura 3.9 se puede ver el foco que es empleado por la vídeo cámara como sistema de iluminación.

La lente de la cámara.

Su correcta elección debe tener en cuenta la distancia de trabajo y el campo de visión necesarios. La utilización de filtros ópticos que resalten los elementos a analizar garantizará el éxito de la tarea. La correcta selección de lentes es importante para alcanzar una solución óptima. Un ejemplo de dicha importancia se puede ver en la Figura 3.10 en la que se muestra la misma imagen capturada con dos tipos de ópticas distintas.

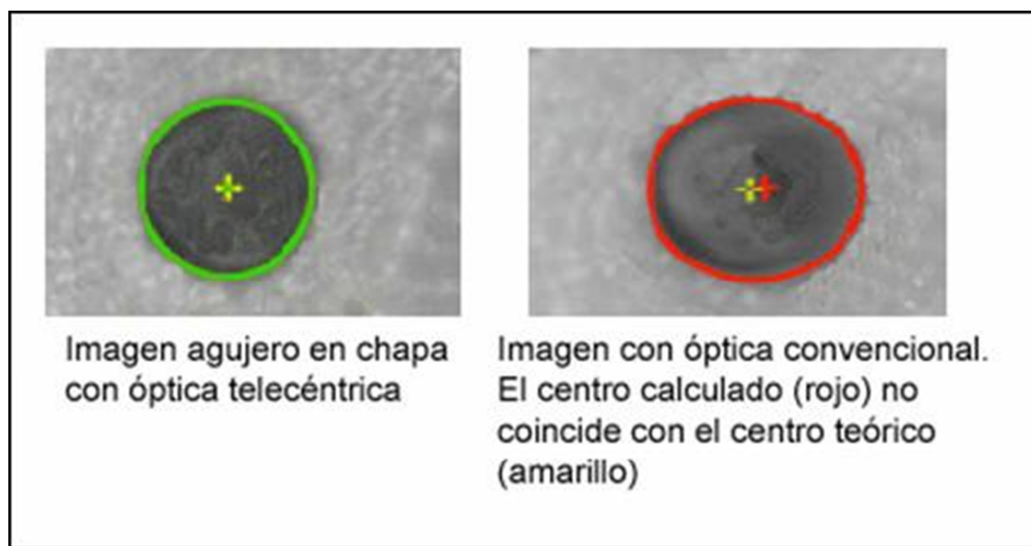


Figura 3.10: Diferencia entre el uso de dos tipos de ópticas

Una o más cámaras para adquirir las imágenes.

Conjunto encargado de recoger las características del objeto en estudio y proporcionar los datos para su procesamiento, por medio de una imagen digital. El tipo de sensor, su tamaño y su resolución deben escogerse en función de los elementos que se desea ver. Las cámaras pueden ser analógicas, pero el precio de las cámaras digitales está disminuyendo, de modo que estas se están usando más a menudo. En la Figura 3.11 podemos ver una cámara de la familia Stingray una serie de cámaras empleadas en la visión artificial con un gran número de funciones de alto nivel.



Figura 3.11: Cámara de alta resolución

Algunos sistemas de visión no utilizan una cámara bidimensional, en su lugar se usa una cámara lineal que produce una sola línea o fila de píxeles. La imagen bidimensional se genera a medida que el objeto pasa bajo la cámara lineal, aprovechando su movimiento, normalmente generado por una cinta transportadora. Uniendo las

distintas filas de píxeles obtenidas a diferentes intervalos de paso, se obtiene una imagen bidimensional.



Figura 3.12: Cámara lineal

En la figura 3.12 vemos un ejemplo de cámara lineal empleada en visión artificial. Se trata de una cámara trilineal de color de hasta 4.000 píxeles de resolución (4096 x 3 píxeles) con salida CameraLink y velocidades de hasta 18.000 líneas por segundo.

Tarjeta de captura o adquisición de imágenes.

Es la interfaz entre el sensor y el ordenador o módulo de proceso que permite al mismo tiempo disponer de la información capturada por el sensor de imagen. La imagen de entrada -una matriz bidimensional de niveles energéticos (por ejemplo, luz)- se divide en elementos de imagen, conocidos como píxeles. Estos forman filas y columnas que abarcan toda la zona de la imagen y representan los niveles de gris en una imagen monocromática o la codificación de color en una imagen en color. Un píxel no puede ser subdividido en regiones de menor nivel de gris o color. Este proceso es un tipo de digitalización espacial. Para cada píxel, la información del nivel de energía también debe ser digitalizada, es decir, los niveles analógicos (variable continua) producidos por la cámara deben ser representados por un número finito de pasos. En muchas aplicaciones es suficiente digitalizar una imagen monocroma con 8 bits por píxel, lo que equivale a 256 pasos, para representar el nivel de gris de cada píxel. En aplicaciones más exigentes puede ser necesario digitalizar a 14 bits (o 16384 niveles). Las imágenes en color son más complejas y pueden ser representados en diferentes formatos. Las imágenes en color normalmente contienen tres veces más información que una imagen monocromática.

Algoritmos de análisis de imagen.

Es la parte “inteligente” del sistema. Su misión consiste en aplicar las necesarias transformaciones y extracciones de información de las imágenes capturadas, con el fin de obtener los resultados para los que haya sido diseñado. Un ejemplo de este elemento constituye el paquete computacional SIVA (Sistema Inteligente de Visión Artificial), y el Image Processing Toolbox (con el cual se ha trabajado en este proyecto) correspondiente del Matlab.

Un procesador de imagen, ordenador o cámara inteligente.

Es el sistema que analiza las imágenes recibidas por el sensor para extraer la información de interés en cada uno de los casos implementando y ejecutando los algoritmos diseñados para la obtención de los objetivos.



Figura 3.13: Cámara inteligente

El procesamiento puede ser realizado por un ordenador u otra opción es utilizar cámaras inteligentes que integran el procesamiento de imágenes dentro de la propia cámara, evitando la necesidad de transferir imágenes a un ordenador externo. La velocidad de proceso de estas cámaras es inferior a la de un ordenador y existen aplicaciones en las que estas no son adecuadas.

En la figura 3.13 podemos ver una cámara inteligente. Esta cámara se suministra con un potente software de desarrollo que permite realizar todo tipo de aplicaciones de visión. Contiene un DSP de Texas Instruments TMS320C64xx a 1GHz y 8000 MIPS. Cuenta con 64MB de Memoria SDRAM y 4 MB de memoria Flash Eprom, conexión Ethernet 100Mbit y Puerto RS-232. Una interfaz para notificar el resultado del análisis o sistema de respuesta en tiempo real.

Con la información extraída se puede mostrar el resultado mediante un monitor, una señal acústica, luminosa, etc., para que un operario realice la tarea correspondiente en función de los datos recibidos o también los sistemas de visión artificial pueden tomar decisiones ellos mismos que afecten al sistema productivo con el fin de mejorar la calidad global de la producción.



Figura 14: Sistema de iluminación

3.1.4.1. EL SISTEMA DE ILUMINACIÓN

Dado que la imagen se forma a partir de la luz que reflejan los objetos, la única posibilidad de obtener una imagen con características constantes en la que se resalten los aspectos de interés y se atenúen los intrascendentes permitiendo o facilitando la resolución del problema es controlando las condiciones de iluminación.

Para ello, se emplea el sistema de iluminación (ver ejemplo en la figura 14), que se encuentra formado por la fuente de luz según entre un determinado esquema de iluminación, lo que en conjunto proporciona unas condiciones de iluminación determinadas.

Hay que tener en cuenta que el problema de iluminación no resulta trivial y su coste supone una parte importante del total del proyecto.

3.1.4.1.1. FUENTES DE LUZ

Entre la multitud de fuentes de luz existentes, las más utilizadas son:

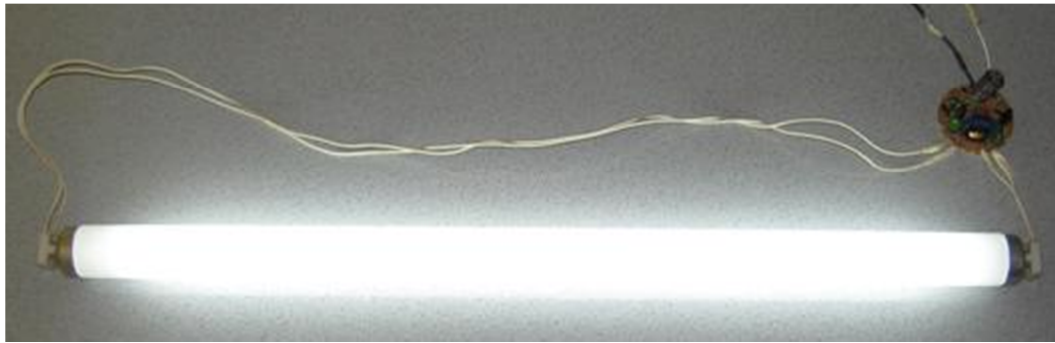


Figura 3.15: Luz fluorescente

Fluorescente de alta frecuencia: no ofrece demasiada luz y presenta deriva con el tiempo, pero su precio económico y su adaptabilidad tanto en forma como en color los hacen atractivos.



Figura 3.16: Luz alógena

Halógena: presenta gran luminosidad, pero desprende calor, su luz es caliente y su precio es caro. Además, presentan envejecimiento.



Figura 3.17: Luz xenón

Xenón: presenta aún mayor luminosidad, aunque los mismos inconvenientes, y un precio aún mayor.



Figura 3.18: Luz LED

LED: admite multitud de configuraciones y están disponibles en multitud de colores, son estables, duraderos, funcionan en baja tensión, aunque su precio es elevado.

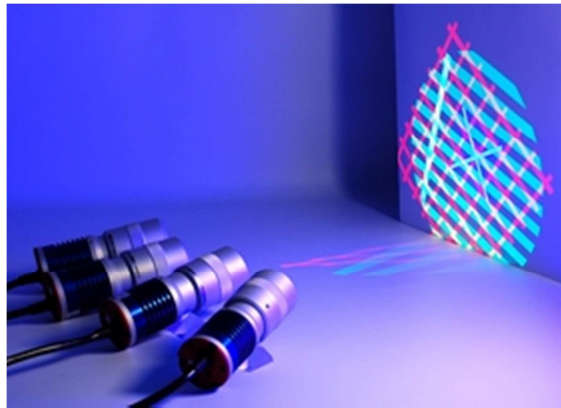


Figura 3.19: Luz laser

Láser: La iluminación mediante láser o luz estructurada se utiliza normalmente para resaltar o determinar la tercera dimensión de un objeto. El método utilizado es colocar la fuente de luz láser en un ángulo conocido con respecto al objeto a iluminar y con respecto a la cámara. Viendo la distorsión de la luz puede interpretarse la profundidad de los objetos a medir.



Figura 3.20: Fibra óptica

Fibra óptica: la iluminación por fibra óptica, es actualmente, la que puede proporcionar la luz más intensa de todos los tipos de iluminación que se utilizan en visión artificial. La idea básica es conducir la luz procedente de una bombilla halógena, o de xenón, que se encuentra en una fuente de iluminación, a través de un haz de fibras ópticas que termina en un adaptador específico para cada tipo de aplicación.

Estos adaptadores pueden tener forma circular, lineal, puntual o de panel, y puede ser de distintos tamaños y dimensiones.

3.1.4.1.2. ESQUEMAS DE ILUMINACIÓN

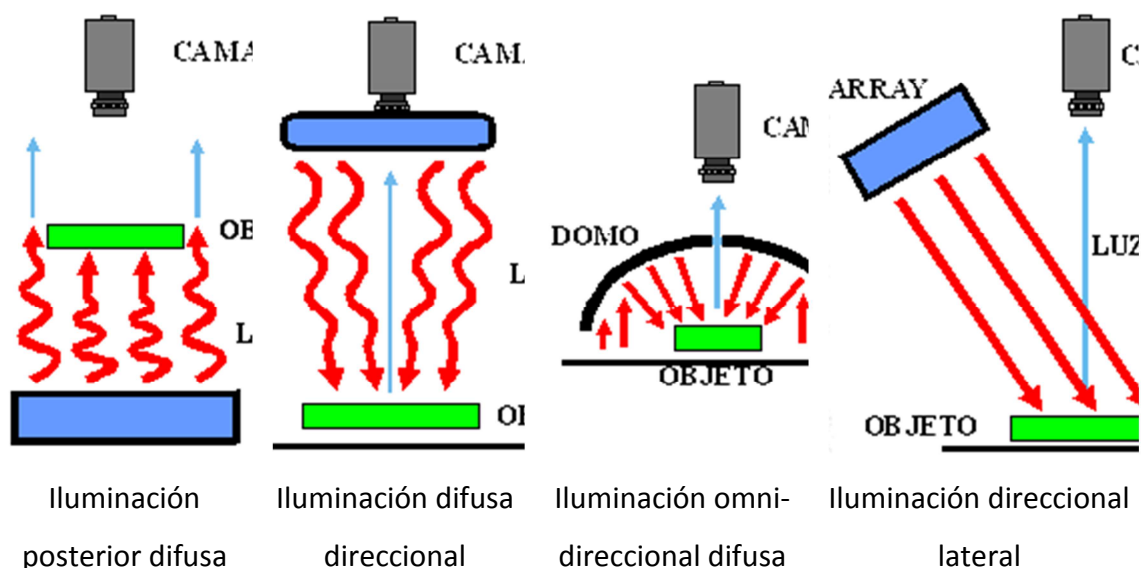


Figura 3.21

Iluminación posterior difusa (Backlight): mediante la iluminación difusa por la parte posterior del objeto se obtiene una imagen de alto contraste, donde la silueta oscura de los objetos se ve resaltada frente al fondo blanco, y por tanto resulta adecuado para medir la forma de los objetos.

Iluminación difusa direccional: emulando a la luz natural pero sin variaciones se emiten los rayos paralelos y en una dirección determinada para lograr una iluminación uniforme.

Iluminación omni-direccional difusa: La cámara consigue una imagen libre de sombras y con gran contraste, gracias a que el objeto es iluminado desde todas direcciones con luz difusa. Para ello se emplea una superficie reflectante de forma semiesférica llamada domo, que actúa como fuente de iluminación al reflejar la luz, con lo que se logra eliminar las sombras y reflejos, y se aumenta el contraste, suavizando las texturas y minimizando la influencia de las rayas, el polvo y los relieves,

así como de las curvaturas que pueda tener el objeto inspeccionado. Se emplea en la detección de marcas de diferentes colores, caracteres y detección de todo lo que suponga un cambio de color tanto en superficies lisas, esféricas, rugosas o con brillo.

Iluminación direccional lateral (Grazing): resalta la textura de los objetos o aquellas características salientes, dado que al incidir la luz con un ángulo muy pequeño sólo será reflejada hacia la cámara cuando se encuentre algún saliente. Por tanto, resulta útil para resaltar protuberancias y hendiduras de la superficie del objeto.

Iluminación con luz polarizada: puede servir para eliminar brillos de objetos brillantes, como en el caso de una bolsa de patatas, dado que los filtros que se instalan en la iluminación y en la cámara impedirán que la luz que no provenga de la fuente de iluminación, y que por tanto presentará distinta polarización, sea capturada por la cámara.

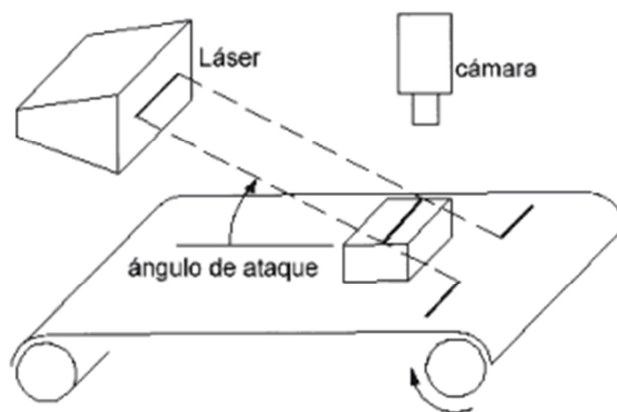


Figura 3.22: Iluminación estructurada

Iluminación estructurada: mediante la proyección de puntos, franjas o rejillas de luz sobre la superficie de trabajo se establece un patrón que permite extraer las características dimensionales del objeto por medio de la medición de la distorsión que sufre dicho patrón de luz ante la presencia de un objeto. Uno de los sistemas más conocidos consiste en la iluminación con un haz láser plano, de modo que al situarse en un ángulo conocido respecto a la cámara, la línea proyectada y distorsionada por la presencia de un objeto presentará un determinado desplazamiento según la profundidad a la que se encuentre el punto, permitiendo de este modo conocer su posición en el espacio, y finalmente reconstruir el objeto 3D mediante técnicas de computación.

3.1.4. 2. LA CÁMARA



Figura 3.23: Cámara de alta velocidad

En la captura de imagen intervienen las cámaras y las tarjetas de adquisición de imagen, estando las cámaras, al igual que el ojo humano, formadas por una lente convergente que proyecta la imagen sobre una superficie sensible a la luz denominada sensor de imagen.

3.1.4. 2.1. SENSORES DIGITALES DE IMAGEN

Los sensores digitales de imagen están formados por una serie de elementos foto sensores (sensibles a la luz) que modifican su señal eléctrica según la intensidad luminosa que reciben, lo que permite la captura de los puntos que conforman la imagen. Estos sensores suelen estar configurados en forma matricial de modo que proporcionan una imagen bidimensional. A nivel industrial también se emplean los sensores con configuración en línea, dada sus altas resoluciones, cuando los objetos se sitúan en superficies en movimiento para aplicaciones de medida (metrología).

Con objeto de reducir costes suelen existir en el mercado multitud de cámaras que emplean lo que se conoce como imagen entrelazada, donde se capturan por un lado las líneas pares y por otro las impares, con lo que al hacerse en instantes de tiempo diferentes no pueden ser empleadas con objetos en movimiento.

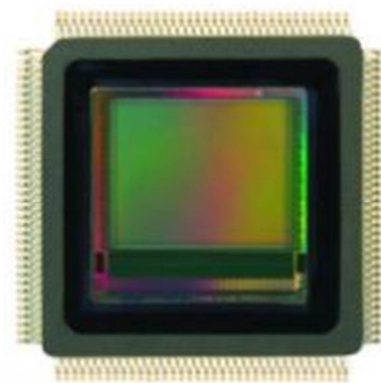


Figura 3.24: Sensor de una cámara digital

A nivel industrial también se emplean los sensores con configuración en línea, dada sus altas resoluciones, cuando los objetos se sitúan en superficies en movimiento para aplicaciones de medida (metrología).

Los sensores son realizados en dos tecnologías, principalmente en CCD y en CMOS.

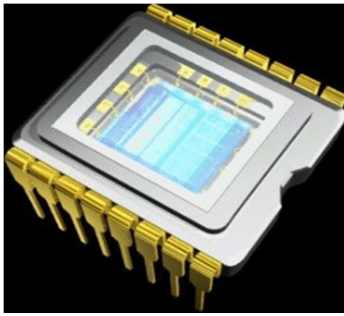


Figura 3.25: Sensor CCD

Sensor CCD

La tecnología CCD (Charge Coupling Device) es la que proporciona una mejor imagen, sobre todo en el caso de las cámaras profesionales en las que se emplea un sensor para cada color, mientras que en las de consumo se emplea un único sensor, de modo que este está constituido como un mosaico de detectores de colores y por tanto captura una menor densidad de puntos. Cuando el sensor ha capturado la imagen envía la información completa serializada fila a fila.



Figura 3.26: Sensor CMOS

Sensor CMOS

La tecnología CMOS es la más económica, y ofrece un menor tamaño, gracias a su grado de integración electrónica. Además, puede capturar partes de la imagen, no teniendo que transmitirla completamente. Al contrario que en los sensores de tecnología CCD, las celdas de la matriz CMOS son totalmente independientes de sus vecinas. La principal diferencia radica en que en estos sensores la digitalización se realiza píxel a píxel dentro del mismo sensor, por lo que la circuitería del sensor es mucho más sencilla.

Características de los sensores.

A la hora de seleccionar una cámara hay que tener en cuenta una serie de características, y entre las principales se destacan:

Resolución: número de píxeles que conforman la imagen capturada.

Sensibilidad: nivel mínimo de iluminación que puede capturar el sensor.

Rango dinámico: margen de luz (intensidad luminosa) que puede capturar el sensor, o sea, diferencia máxima entre luz y oscuridad que puede distinguir.

Señal/ruido y Smear: influencia entre píxeles.

Velocidad: velocidad máxima a la que puede capturar imágenes, siendo su medida en frames por segundo [fps].

3.1.4. 2.2. OBJETIVOS

El objetivo tiene como función concentrar los rayos de luz, provenientes de los objetos a visualizar, en un plano donde se formará la imagen para ser captada por el sensor. Está formado por el diafragma, que controla la apertura y por tanto limita el haz de rayos entrante, la lente, que enfoca el objeto, y el zoom, que permite variar la distancia focal (distancia al objeto).



Figura 3.27: Objetivo de alta resolución

El diafragma permite regular la apertura del objetivo y con ello la luminosidad de la imagen. Además, aunque al cerrar el diafragma los rayos salen enfocados independientemente de la distancia al objeto, entrará menos luz y por tanto la escena será más oscura, con lo que se requerirá un mayor tiempo de exposición.

La lente viene caracterizada por la distancia focal (f), que es la distancia desde su centro hasta el punto en el que convergen los rayos que llegan paralelos a ella. Así, el aumento de la distancia focal disminuye el campo visible y con ello la distancia de trabajo de la misma.



Figura 3.28:Objetivos telecéntricos

De este modo, el desenfoque se produce según el objeto se aleja o acerca de la posición ideal, dado que la imagen dejará de formarse sobre el sensor, y por tanto cada punto de la imagen no se corresponderá con su elemento correspondiente del sensor. Para corregirlo, se cambia la distancia focal de la lente, lo que se traduce en variar la distancia entre lente y sensor.

Existen una clase de objetivos, llamados telecéntricos, que permiten ver los objetos en proyección ortográfica; siempre y cuando sean más pequeños que el diámetro de la propia lente.

Características de los objetivos.

Campo visible (FOV): es el área visible por la lente para una cierta distancia de trabajo en la que los objetivos aparecen enfocados.

Profundidad de campo (DOF): es el rango de distancias a las cuales los objetos se encuentran enfocados.

3.1.5 CÓMO FUNCIONA LA VISIÓN ARTIFICIAL

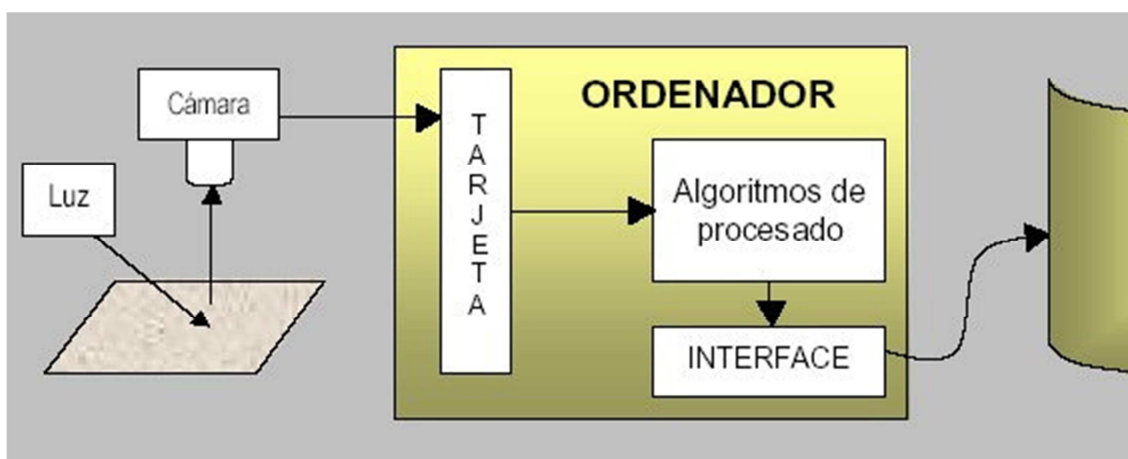


Figura 3.29: Elementos empleados en una aplicación de visión artificial

Aunque cada aplicación de visión artificial tiene sus especificidades, se puede decir que existe un tronco común de etapas entre ellas. No necesariamente debe cubrirse todas en una implementación concreta. Hay algunas veces que sólo se tiene un subconjunto de las fases que se van a citar. Por otro lado, aunque la exposición muestra un encadenamiento temporal de una etapa sobre otra, no es real esta simplificación; se hace para facilitar la comprensión y en la puesta en práctica siempre se encuentra realimentación entre las distintas fases.

Los dos pilares del sistema físico de visión artificial son: el sistema de formación de las imágenes y el sistema de procesamiento de éstas. En el primer apartado estaría constituido por el subsistema de iluminación, de captación de la imagen y de adquisición de la señal en el computador. Una vez introducida la señal en el computador, ésta es procesada mediante los algoritmos para transformarla en información de alto nivel. La cual puede ser utilizada para su representación visual, para actuar en el planificador de un robot o ser fuente de datos para un autómata programable. En definitiva, múltiples periféricos pueden ser receptores de esta información y vincularse con el sistema de procesamiento de las imágenes.

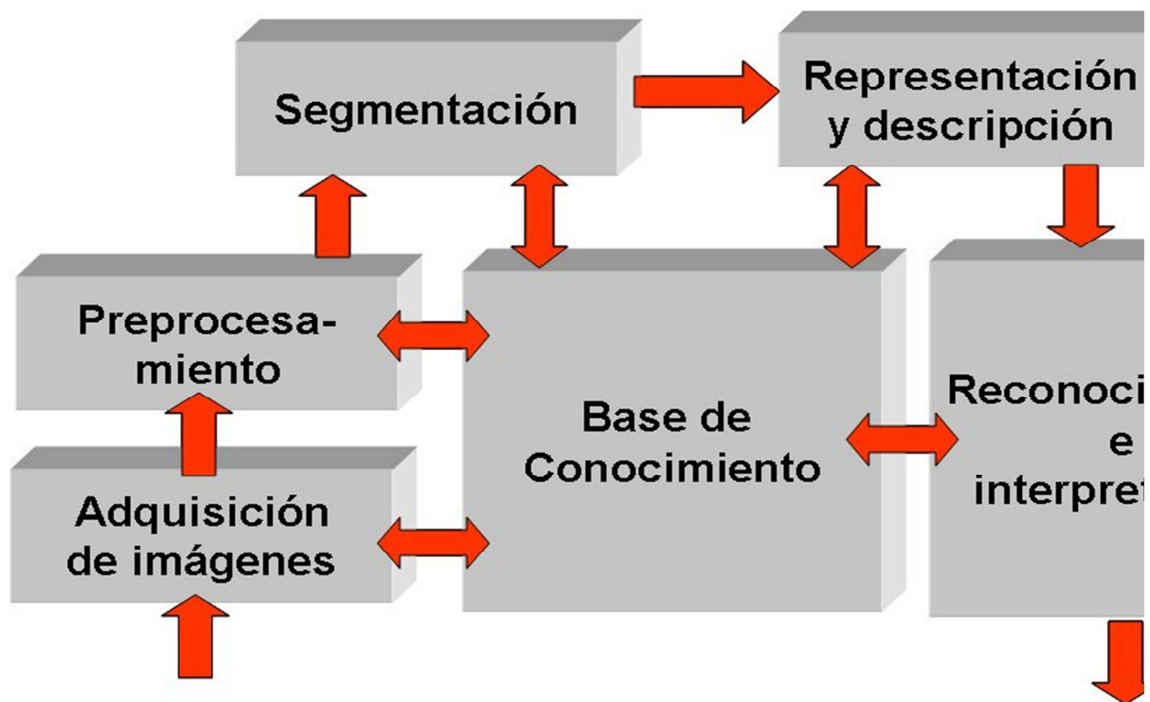


Figura 3.30: Etapas de una aplicación de visión artificial

3.1.5.1. ADQUISICIÓN DE LA IMAGEN

La primera etapa es la construcción del sistema de formación de las imágenes. Su objetivo es realzar, mediante técnicas fotográficas (iluminación, óptica, cámaras, filtros, pantallas,...), las características visuales de los objetos (formas, texturas, colores, sombras,...). El éxito de muchas aplicaciones depende de un buen diseño en esta primera etapa.

3.1.5.2. PREPROCESADO

Una vez adquirida la imagen se pasará a la etapa de preprocesado. El objetivo es mejorar la calidad informativa de la imagen adquirida. Se incluyen operaciones de mejora de la relación señal-ruido (denoising), SNR, de atenuar las imperfecciones de la adquisición debido a la función de transferencia del sistema de captación de imágenes (deconvolution), de regularizar la imagen, de mejorar el contraste o de optimizar la distribución de la intensidad (enhancement) o de realzar algunas características de la imagen, como bordes o áreas.

3.1.5.3. SEGMENTACIÓN

En esta fase se divide la imagen en áreas con significado. Se trata de decidir qué partes de la imagen necesitan interpretación o análisis y cuáles no. A menudo es posible y necesario refinar la segmentación. Por ejemplo, si la aplicación debe encontrar arañazos o grietas en un vidrio, la primera fase de segmentación suele encontrar elementos que son las propias grietas o arañazos además de suciedad y polvo. Otro ejemplo, en una imagen de satélite se determina las zonas de agua, de cultivo, urbanas, carreteras, ... A menudo se perfecciona este resultado para eliminar de la imagen la suciedad y el polvo de los elementos a analizar.

Existen varias técnicas: umbralizaciones, discontinuidades, crecimiento de regiones, uso del color o de movimiento, etc.

3.1.5.4. REPRESENTACIÓN Y DESCRIPCIÓN

Una vez dividida la imagen en zonas con características de más alto nivel se pasará a su extracción de las características. Básicamente son de tipo morfológico, tales como área, perímetro, excentricidad, momentos de inercia, esqueletos, pero también se puede emplear características basadas en la textura o en el color.

Fíjese que se ha pasado de una información visual primaria a algo más elaborado. Con las características analizadas ahora cada región se debe de clasificar e interpretar. Por tanto, se diseñarán clasificadores que le dé a cada área segmentada una etiqueta de alto nivel, como por ejemplo, en una imagen aérea qué zonas son tierras de cultivo, áreas urbanas, etc. Existe un elenco de técnicas de clasificación, como redes neuronales, sistemas expertos, lógica difusa, clasificadores estadísticos, etc.

3.1.5.5. RECONOCIMIENTO E INTERPRETACIÓN

Una vez que la imagen ha sido satisfactoriamente segmentada y han sido extraídas las características, el procesador puede hacer una serie de pruebas y mediciones en los elementos de interés que aparecen en la escena. Existen multitud de algoritmos que

nos ayudan a obtener los resultados deseados. Entre ellos, cabe destacar sus funcionalidades:

- Detección de formas
- Análisis geométrico
- Comparación con patrones
- Medición de objetos
- Detección de objetos difusos

Estos algoritmos, usados estratégicamente, nos permiten decidir si una pieza se ha fabricado correctamente o no, guiar un robot hacia un objeto concreto o reajustar la maquinaria de producción automáticamente.

3.1.6. JUSTIFICACIÓN DE LA INVERSIÓN EN VISIÓN ARTIFICIAL

En las industrias productivas la visión artificial se está convirtiendo rápidamente en un factor clave en el desarrollo de la calidad total dentro de los diferentes procesos de automatización industrial. Su implementación en una empresa genera un aumento en el nivel de producción y una reducción en los costos de fabricación, elevando los niveles de competitividad en el mercado nacional e internacional. La visión artificial permite inspeccionar el proceso de producción sin fatigas ni distracciones, facilitando la cuantificación de las variables de calidad traduciéndose en un mejoramiento continuo.

Las inspecciones realizadas por los seres humanos, a menudo no pueden cumplir con los requisitos de la industria moderna respecto a la velocidad de producción, calidad de producto y costes de producción. Los humanos se cansan, cometen errores y los criterios que se aplican durante las inspecciones son inevitablemente subjetivos. En algunos casos, no es humanamente posible llevar a cabo las tareas de inspección debido a las condiciones ambientales. Las cámaras y el resto de componentes de un sistema de visión artificial, por el contrario, llevan a cabo las mediciones con una precisión constante y a un ritmo que es establecido por el propio proceso de

producción. Estas ventajas han llevado a una creciente aceptación de la visión por industrias de todo el mundo.

3.1.6.1. COSTES DE LOS MATERIALES

En la mayoría de las aplicaciones, evitar la producción de piezas defectuosas mediante el uso de un sistema de visión industrial tendrá un período de amortización muy corto. Para evitar que se fabriquen piezas defectuosas, el sistema de inspección automática, ya sea muestreando el 100% en la línea de producción o bien usado fuera de línea tomando muestras, debe formar parte del control estadístico de procesos (SPC) del sistema productivo. Esto significa que el sistema indica cuando un parámetro de control deriva hacia el límite de tolerancia, o es simplemente demasiado errático. El sistema de visión puede tomar medidas correctivas antes de que el límite sea superado.

3.1.6.2. COSTES DE LA MANO DE OBRA

La reducción de la mano de obra es también un importante ahorro de costes, ya que muchas de las tareas realizadas por la visión industrial pueden sustituir a personas directamente. Además, deben ser considerados los ahorros en selección de personal, prestaciones sociales y los aumentos salariales anuales.

3.1.6.3. COSTES DE LA CALIDAD

La creciente conciencia del costo de la calidad a través de las normas ISO9001, significa que el uso de la visión artificial puede ofrecer un estándar más objetivo, fiable y consistente en la inspección de productos.

El ahorro en la optimización del uso de materiales, seguimiento de la calidad de los proveedores y garantía de calidad de los productos acabados pueden llevar a ahorros tanto tangibles como intangibles. El costo de los trabajos de reparación en garantía se puede reducir y además se mejora la confianza de los clientes consiguiendo pedidos recurrentes y una mayor cuota de mercado.

3.2. LA VISIÓN ARTIFICIAL EN LOS VEHÍCULOS

Ahora estamos viendo los primeros anuncios de coches en los que se destaca el uso de esta tecnología. Varios modelos incluyen ya el reconocimiento de señales y son capaces de adaptar la velocidad del coche a la indicada por la señal. Combinando esta tecnología con sensores de radiofrecuencia permite a los coches es aparcamiento automático sin la intervención de conductor. Otro uso interesante es el reconocimiento de los carriles, que avisa al conductor si se está saliendo involuntariamente de este.

En los espejos retrovisores siempre hay un punto muerto en el que no vemos el coche que viene por detrás, pues bien, instalando una cámara en la parte trasera del coche, este nos avisa cuando un coche se encuentra en nuestro punto muerto de visión, ayudándonos a no tener ningún accidente.

En cuanto a los vehículos de dos ruedas no he encontrado ningún estudio llevado a cabo, para la implementación de la visión artificial.



Figura 3.31

3.3. PRIMERAS PRUEBAS CON OJOS BIÓNICOS

Hago mención a este estudio ya que me ha parecido muy interesante, aunque no se aplique la visión artificial en vehículos que es de lo que trata el proyecto.



Figura 3.32 Ojo biónico

Uno de los grandes retos a los que se enfrentan los científicos es poder curar algunas de las enfermedades vinculadas a la visión, es decir, curar la ceguera. Hace dos años un equipo de científicos australianos presentó el primer ojo biónico, un dispositivo capaz de excitar, de manera artificial, el ojo de un paciente que esté perdiendo de manera progresiva la visión por enfermedades degenerativas de la retina. Tras realizar diversas pruebas, el equipo ha anunciado su siguiente gran paso: **comenzar** pruebas con pacientes reales el próximo año.

El equipo de la New South Graduate School of Biomedical Engineering ha desarrollado ya dos prototipos de este sistema de visión artificial que podría servir para restaurar la visión en pacientes que sufran de enfermedades degenerativas de la retina. La investigación ha dado como resultado un sistema que mediante la colocación de 98 electrodos en la retina y estimular sus células nerviosas que son las que convierten la luz captada por el ojo en señales eléctricas, es decir, gracias a este sistema alguien que

sufre pérdida de visión al menos podría distinguir ciertos patrones y formas. En el prototipo que han diseñado, se coloca una cámara en la montura de unas gafas, en una implementación bastante discreta, que se conectan a una unidad de procesamiento externo y éste, a su vez, se conecta al chip cuyos electrodos se conectarán al ojo del paciente. Con este esquema, solamente queda un detalle para completar el prototipo, el equipo tiene que encapsular el sistema que se va implantar en el paciente y para ello deben buscar materiales que sean resistentes y biocompatibles.

Con la idea de desarrollar el encapsulado y, por tanto, desarrollar por completo el prototipo y además mejorar la capacidad del laboratorio para realizar nuevas investigaciones, el equipo ha invertido 2,5 millones de dólares (1,87 millones de euros) en instrumental y equipamiento que les permitirán abrir nuevas líneas de investigación relacionadas con el desarrollo de implantes biomédicos.

4. PLANTEAMIENTO Y MEDIOS UTILIZADOS

4.1. PLANTEAMIENTO

Como hemos explicado anteriormente en el apartado de objetivos, dividiremos el proyecto en dos partes diferenciadas; en la primera desarrollaremos un sistema de control de dirección para un vehículo estable, basado en el seguimiento de una línea negra sobre un fondo uniforme, en nuestro caso de color blanco. La segunda parte consistirá en calcular la inclinación que lleva el vehículo en todo momento y así obtener una señal que posteriormente podría ser incluida en un sistema de control.

El vehículo con el que vamos a hacer el proyecto es una "moto" de Lego. Aunque le llamamos "moto", en la primera parte del proyecto llevará dos ruedas traseras para mantenerla estable y por lo tanto sería más bien un triciclo.

La idea para desarrollar el control de dirección es instalar en la moto una cámara de video, que vaya captando en todo momento la línea que debe seguir la moto. Para ello, la cámara tendrá que estar situada en la parte delantera de esta. Cada imagen que capte la cámara queremos que sea transmitida al ordenador, donde la procesaremos y obtendremos una serie de valores, que serán enviados a la CPU de la moto. La CPU debe recibir estos valores y activar los actuadores que lleva conectados. Estos actuadores controlarán el movimiento de la moto. Este ciclo de control permitirá a la moto avanzar siguiendo la línea negra.

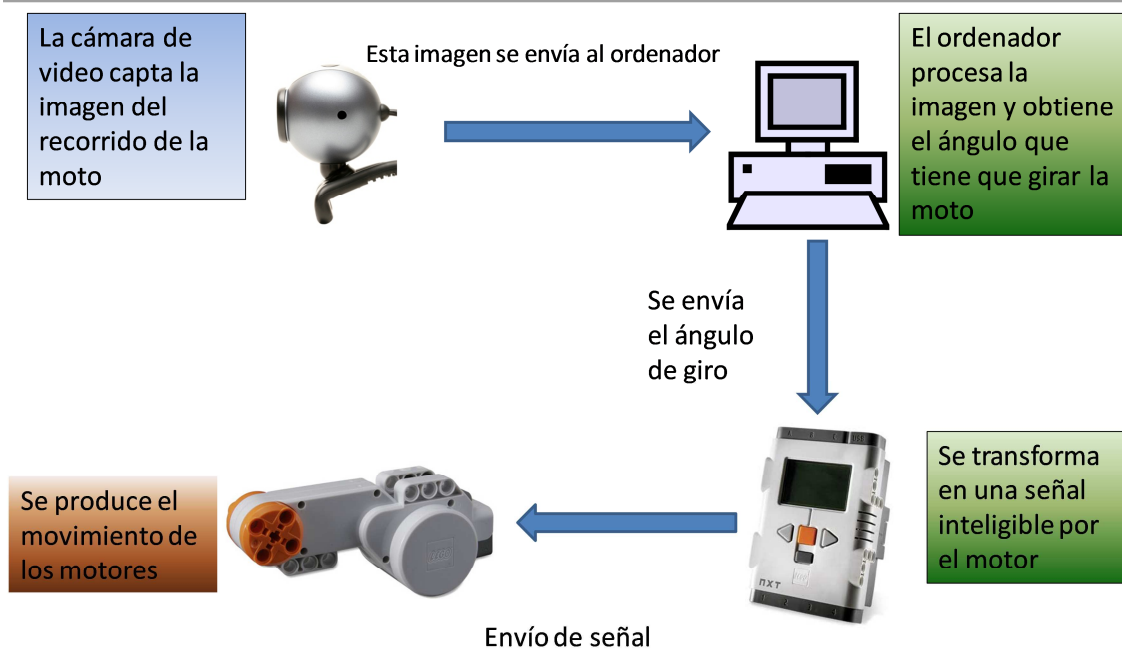


Figura 4.1

Para el cálculo de la inclinación, en un principio, teníamos pensado obtenerla sin añadir ningún otro dispositivo a la moto, solamente con la imagen captada por la cámara, pero como se explicará posteriormente, se vio que no era posible y se instalaron dos punteros láser adicionales que proporcionan las referencias para el cálculo de la inclinación.

4.2. MEDIOS UTILIZADOS

Para tener una mejor visión global de este proyecto y favorecer a la comprensión del mismo, detallaremos el hardware y software utilizados.

4.2.1. HARDWARE

El hardware, son todas las partes tangibles, físicas, que pertenecen al sistema.

4.2.1.1. LEGO MINDSTORM NXT

Lego Mindstorm está catalogado como un juego para el aprendizaje de los principios básicos de la robótica, construido por Lego. En principio este juego está orientado hacia niños, o para su uso en escuelas e institutos, ya que posee elementos básicos de

la robótica como el montaje de piezas y la programación básica de acciones de forma interactiva, con un software de intuitivo y de fácil manejo.

Pero en realidad se le puede sacar mucho más partido del que aparenta. Limitado únicamente por la interfaz de Lego, el sistema se compone de una serie de sensores y actuadores de fácil conexión con un procesador; que programados con libertad, es el hardware perfecto para la implementación de nuestro proyecto. Para ello, aproveché el trabajo desarrollado por la gente de la Universidad de Aachen (Alemania), que han conseguido comunicar la CPU del Lego Mindstorms con Matlab, y han creado una serie de funciones muy útiles para poder programar este dispositivo.

Para este proyecto, compramos una caja de Lego Mindstorm NXT, que contiene:

- La CPU o “Brick” de Lego, que es un bloque de 11cm de largo y 4cm de grosor, que contiene una pantallita LCD en blanco y negro, una serie de botones para su manejo, y puertos de conexión para los actuadores y los sensores.
- Un sensor infrarrojo, uno de posición y uno de sonido.
- Tres motores paso a paso.
- Cables de conexión para conectar los actuadores y sensores con la CPU.
- Piezas para el montaje de diversos robots de Lego.
- Cargador de la batería.

Para su construcción me ayudé del libro **“Building robots with Lego Mindstorms NXT”**, con diferentes tipos de montajes que se podían hacer con la caja de Lego Mindstorm NXT, en el cual estaba incluido el montaje de una moto. Al final este modelo resultó ser muy inestable y endeble, así que diseñe un montaje que fuese óptimo para los requerimientos del proyecto.

La moto está compuesta por dos motores paso a paso ,situados en la parte trasera, utilizados para la tracción, que se controlan de forma simultánea en Matlab y un motor idéntico a los anteriores para el control de la dirección, situado en la parte delantera de la moto, como se puede ver en las imágenes. [\(numerar imágenes\)](#)

La CPU, que es el elemento más pesado, está situada lo más cerca posible del suelo, para bajar el punto de gravedad y que la moto sea más estable, además de actuar como parte del chasis, sirviendo como unión de las piezas de la parte trasera y delantera de la moto.

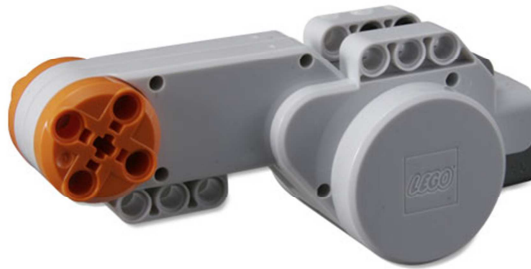


Figura 4.2

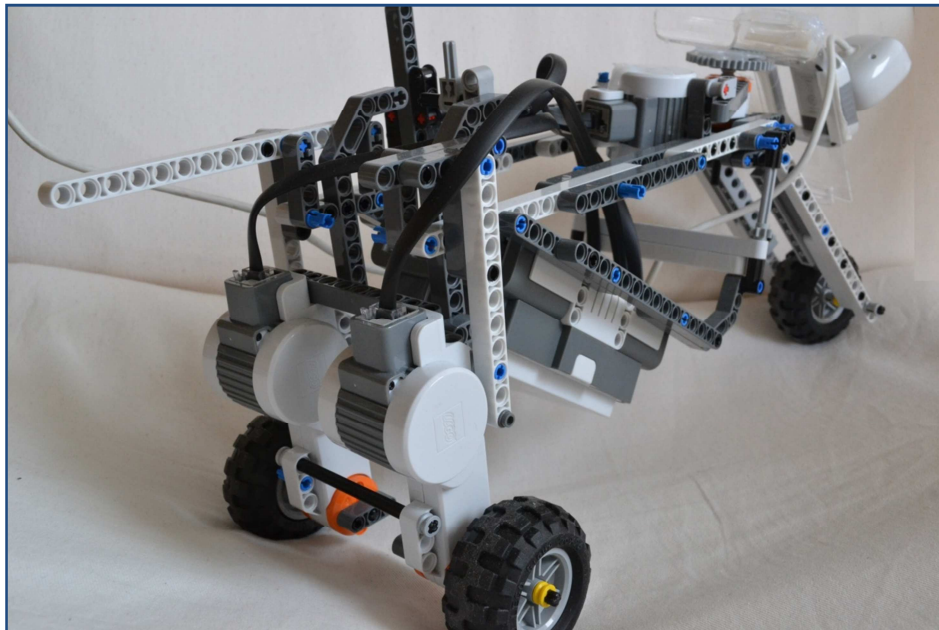


Figura 4.3

4.2.1.2. Microsoft LifeCam VX-800

Es una cámara web con micrófono integrado fabricada por Microsoft, que va conectada y alimentada mediante el puerto “High Speed-USB”. La calidad de imagen que capta son 640x480 píxeles.



Figura 4.4

La cámara va situada en la parte delantera de la moto, sobre un soporte que gira sincronamente con la dirección de la moto. Este soporte está inclinado 60 grados sobre la horizontal de la moto, lo que hace que la cámara apunte hacia el suelo, captando el recorrido que deberá seguir la moto. A esta posición final de la cámara se llegó después de varias pruebas, de forma que se consiguiera la correcta visualización de la línea negra. Inicialmente la cámara no giraba sincronamente con la dirección ya que estaba anclada al chasis de la moto, lo que hacía que en el momento de giro la cámara perdiese la visión de la línea negra, quedando enfocada hacia el frente y no en la dirección en la que gira la moto.

Las imágenes captadas por la cámara se envían por el puerto USB y son procesadas por el programa de control en Matlab.

En la siguiente imagen se ve como el soporte de la cámara va anclado en la dirección.



Figura 4.5

4.2.1.3. PUNTEROS LASER

Estos dos punteros láser nos sirven para determinar la inclinación que lleva la moto en todo momento. Los punteros laser van colocados por debajo de la cámara, en el mismo plano y enfocando hacia el suelo, con 60° de inclinación sobre la horizontal de la moto, situándose las proyecciones de los punteros en la región que capta la cámara.

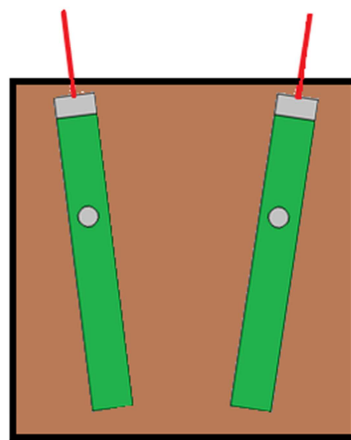


Figura 4.6

4.2.1.4. MODELO DE PRUEBAS DE LA MOTO

Para realizar todas las pruebas previas al montaje final, he fabricado un modelo en cartón de la moto. Al ser menos pesado y menos frágil que la moto de Lego, lo he podido manejar, girar e inclinar sin problemas, así como cambiar la distribución de la cámara y los punteros, lo que me ha permitido realizar múltiples ensayos de forma sencilla y ayudarme a identificar la colocación óptima de la cámara y los punteros láser en la moto real.



Figura 4.7

4.2.1.5. DISPOSITIVO BLUETOOTH

La tecnología inalámbrica Bluetooth es un medio de comunicación mediante ondas de radio de corto alcance (2.4 GHz), que sirve para poder comunicar y sincronizar dispositivos informáticos en distancias de hasta unos 10 metros, incluso con obstáculos de por medio.

El "brick" de Lego lleva integrado un dispositivo de comunicación Bluetooth. Para poder comunicarnos con él desde el ordenador, hemos instalado el dispositivo de comunicación vía Bluetooth, que se muestra en la imagen.



Figura 4.8

4.2.2. SOFTWARE

Llamamos software, al conjunto de componentes lógicos necesarios que hacen posible la realización de tareas específicas. Esto incluye los programas informáticos que usamos a diario en nuestro ordenador (Word, Excel, Acrobat Reader etc.), así como el propio sistema operativo del ordenador (Windows, Linux, IOS), los drivers que hacen que funcione el hardware en nuestro ordenador etc.

4.2.2.1. MATLAB

Matlab (abreviatura de **Matrix Laboratory**) creado por Cleve Moler en 1984, es un lenguaje de programación de alto nivel, que desarrolla un entorno interactivo para visualización de datos, desarrollo de algoritmos, análisis de datos y cálculo numérico.

Nos da la posibilidad de resolver problemas matemáticos y técnicos con mayor rapidez que otros lenguajes de programación como C, C++ o FORTRAN.

Matlab está diseñado para trabajar en muchos campos diferentes como el procesamiento de imágenes y de señales, comunicaciones, sistemas de control, sistemas de prueba y medición etc. Para cada uno de ellos ofrece un paquete de herramientas, que incluyen multitud de funciones predeterminadas que simplifican muchísimo la labor del programador.

De todos estos paquetes de herramientas, el que me ha sido de gran ayuda y he utilizado en todo momento ha sido el "Image Acquisition Toolbox" (Paquete de Herramientas de Adquisición de Imágenes). Este paquete permite adquirir directamente imagen y video de cámaras web, o cámaras de video o de fotos conectadas al ordenador y son procesadas al momento en Matlab o Simulink (Simulink es un entorno de programación visual dentro del Matlab; su mayor aplicación es el diseño de modelos de control). Con este paquete se puede detectar directamente el hardware que tenemos conectado y configurarlo desde el Matlab a nuestro gusto, por ejemplo cambiando la calidad que queremos recibir de la imagen (640x480, 320x240 píxeles), el número de imágenes que queremos que capte, el modo de disparo, cuántas imágenes por disparo etc. Tiene capacidad de reconocimiento para casi todo tipo de cámaras y fabricantes, desde la más baratas webcams hasta cámaras de video profesional.

Para el control de la CPU de Lego mediante Matlab también he utilizado un paquete de herramientas, que no ha sido desarrollado por Matlab, sino por un grupo de estudiantes de la Universidad de Aachen (Alemania). El paquete se llama *"RWTH - Mindstorms NXT Toolbox for MATLAB"* y ha sido diseñado para controlar los robots de Lego Minstorm NXT desde Matlab mediante la conexión inalámbrica Bluetooth o por el cable USB. La ventaja principal que tenemos al poder programar un robot desde Matlab es que podemos combinar simples funciones de robótica con complejas funciones matemáticas y de visualización.

La instalación de este paquete en Matlab es muy simple, hay que descargar de la página la carpeta con las funciones y agregarlas a la dirección donde tengas el resto de funciones o programas creados con los que estés trabajando.

Adaptar la CPU de Lego al Matlab es algo más complicado. Primero se descarga el driver que hace que nuestro ordenador reconozca el "brick". Una vez que el ordenador lo reconoce como un dispositivo, se aplica un programa llamado NeXTEplorer proporcionado también por la Universidad de Aachen, que nos permite visualizar y modificar el firmware de Lego, así como el contenido interno del "brick". Una vez abierto el programa copiaremos a la raíz de la memoria del dispositivo el archivo MotorControl.rxe, el cual transforma las instrucciones que nosotros le pasamos desde el Matlab a un lenguaje entendible por el dispositivo de Lego. Aquí, ya estamos preparados para poder conectar la CPU de Lego mediante Bluetooth con el ordenador.

5. IMPLEMENTACIÓN

5.1. CONTROL DE DIRECCIÓN

El siguiente flujograma muestra la visión general del control de dirección, posteriormente se detallará cada sistema.

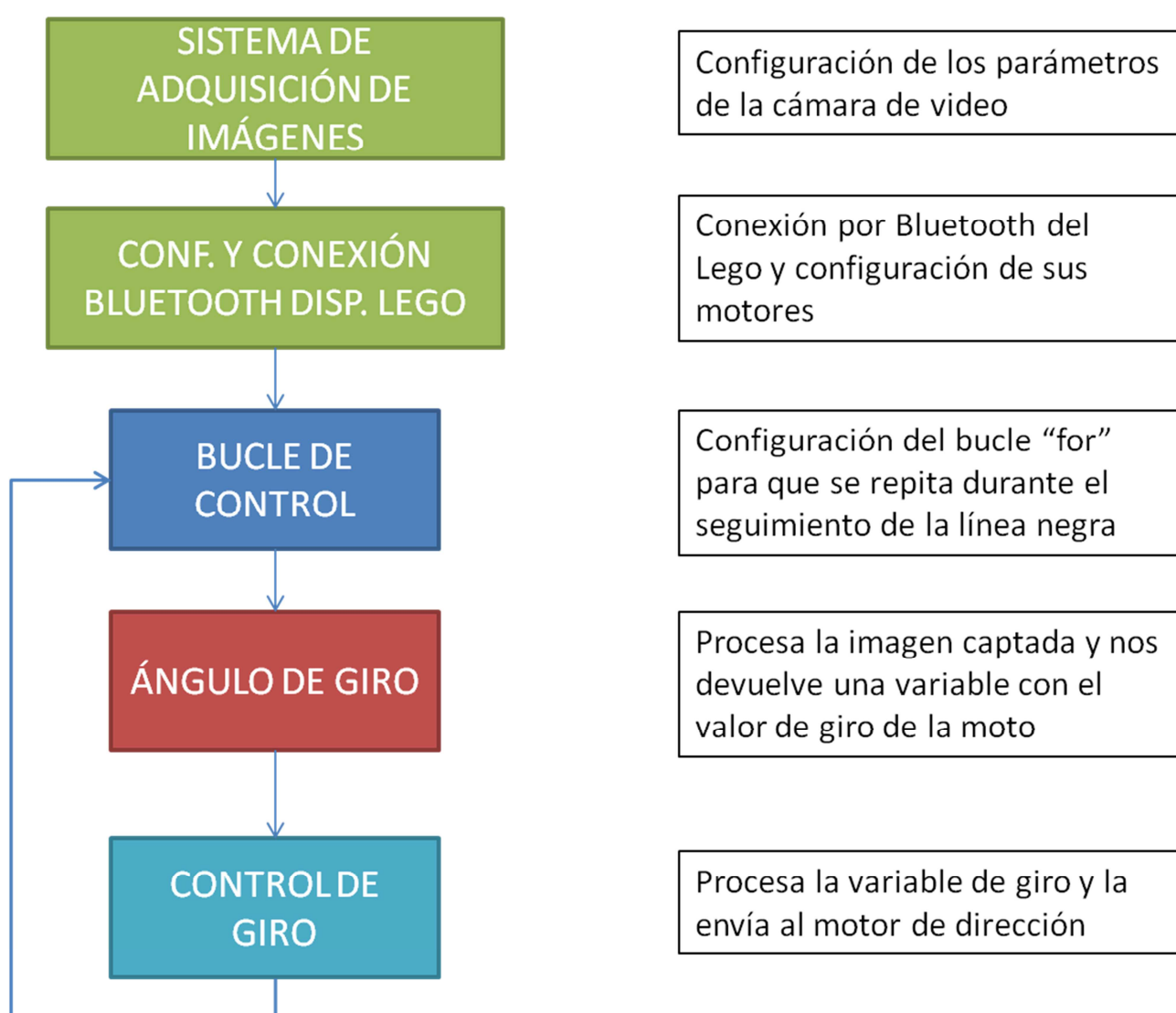


Figura 5.1

5.1.1. SISTEMA DE ADQUISICIÓN DE IMÁGENES

Los videos y las películas que estamos acostumbrados a ver son una secuencia de imágenes visualizadas a una frecuencia determinada, que hace que visto por el ojo humano parezca una reproducción continua. En este caso el funcionamiento es similar, nuestra cámara va captando varias imágenes por segundo y nosotros tenemos que determinar la frecuencia de adquisición de esas imágenes. El paquete de herramientas “*Image Acquisition Toolbox*” que nos proporciona el Matlab nos da la oportunidad de configurar parámetros como la calidad que queremos recibir de la imagen (640x480, 320x240 píxeles), el número de imágenes que queremos que capte, el modo de disparo, cuántas imágenes por disparo etc.

Lo primero que se hace es configurar el puerto donde está conectada nuestra cámara para que el Matlab la reconozca, en nuestro caso tendremos conectada la cámara a la entrada 1. La cámara permite capturar imágenes en dos resoluciones 640x480 y 320x240 píxeles, nosotros optamos por la de mayor resolución, ya que nos permitirá un mejor visionado de la línea negra y de los punteros láser.

Ahora tenemos que definir el número de imágenes que queremos que capte la cámara en cada disparo. Como aquí se analiza una única imagen en cada ciclo de control, tomamos una imagen por cada captura. También hay que definir el número de disparos; nosotros lo dejaremos en infinito ya que finalmente irá determinado por el número de veces que queramos ejecutar el bucle de control. Por último, fijamos que las imágenes captadas lo sean en el espacio de colores RGB (en inglés rojo, verde y azul). La imagen capturada será una composición de tres matrices, cada una correspondiente a un color.

De esta forma tenemos ya configurada la cámara para devolver la imagen en el formato deseado.

5.1.2. CONFIGURACIÓN Y CONEXIÓN DEL "BRICK" DE LEGO

Para conectar el "brick" de Lego y el ordenador mediante Bluetooth, hay que definir un archivo en Matlab llamado "bluetooth.ini" que contiene toda la información de la configuración del dispositivo Bluetooth. Una vez definido este archivo se procede a ejecutar la función de conexión. Una vez conectado, mandaremos la configuración inicial a los motores, en donde queda definida la velocidad, el método de frenado, de acelerado etc.

5.1.3. BUCLE DE CONTROL

Se ha definido un bucle "for" que se repite durante lo que dure el seguimiento de la línea negra. Dentro del bucle se llamará al siguiente programa, el cual es el más importante, ya que nos devuelve la variable del giro de la moto.

5.1.3.1. PROGRAMA ÁNGULO DE GIRO

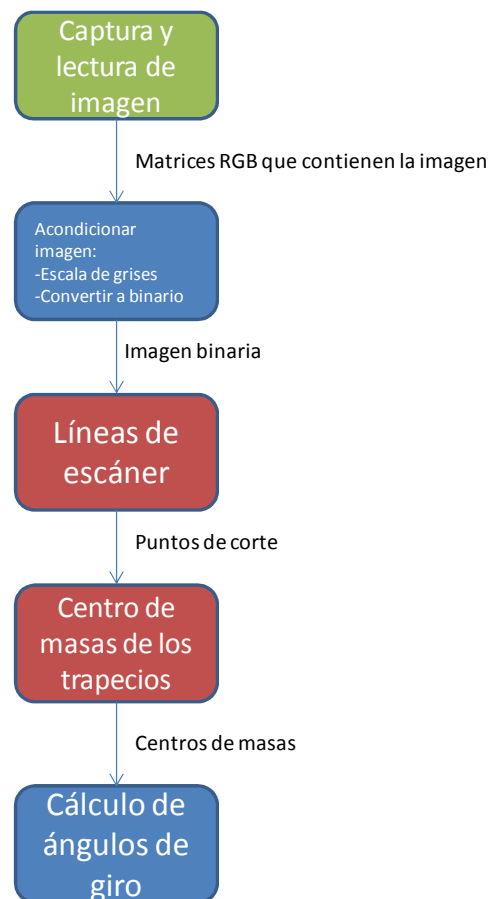


Figura 5.2

La implementación del sistema de dirección en la moto, se basa en el seguimiento de una línea negra, que en este caso es una cinta aislante de este color sobre una plantilla blanca, compuesta por varios folios pegados entre sí.

El primer desarrollo de este programa de seguimiento se basaba en los procesos que se detallan a continuación. Las variables y nomenclatura se definen en la imagen XX:

1. Se lee la imagen de la línea negra proveniente de la cámara en Matlab, se convierte a escala de grises y posteriormente a binario (0=negro, 1=blanco).
2. Se establecen líneas horizontales de escáner en las que se cuenta y detecta el color de los píxeles que contienen. Con esta información se determina el ancho de la línea negra y se calcula su punto medio.
3. Conocidas las posiciones de estos puntos medios y las distancias entre ellos, se obtiene la trayectoria de la línea negra.
4. Conocida la altura y ángulo de instalación de la cámara y la velocidad de la moto, se calcula el ángulo de giro del motor de dirección requerido para que la moto gire siguiendo la línea negra.

La siguiente imagen detalla el proceso descrito anteriormente

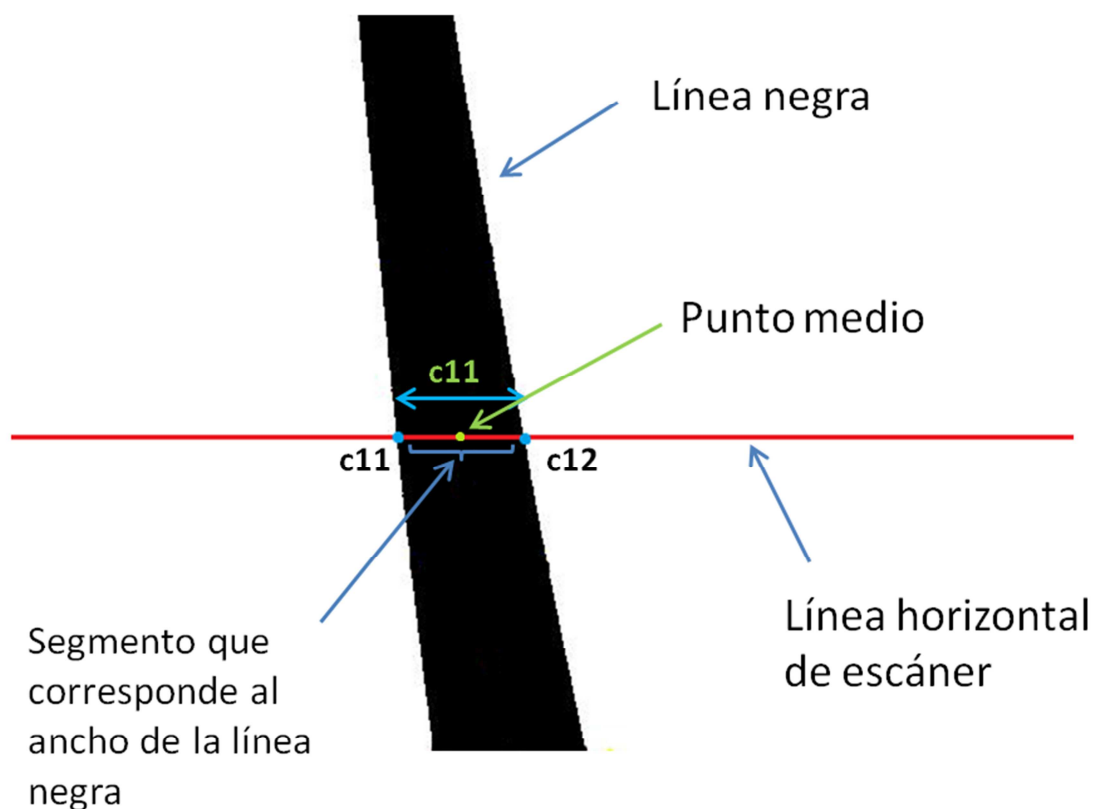


Figura 5.3

El siguiente flujograma detalla el proceso de cálculo de los pasos 1 y 2 anteriores en los que se calculan los segmentos y sus puntos medios para cada línea horizontal.

Diagrama de flujo:
Líneas de escáner

Pixel negro=0
Pixel blanco=1

n1: cuenta el
número de píxeles
negros seguidos

c1(x): vector que guarda en cada
posición el número seguido de
píxeles negros escaneados

pm1: punto medio de la línea
negra

c11: punto del extremo izquierdo
de la línea negra

c12: punto del extremo derecho
de la línea negra

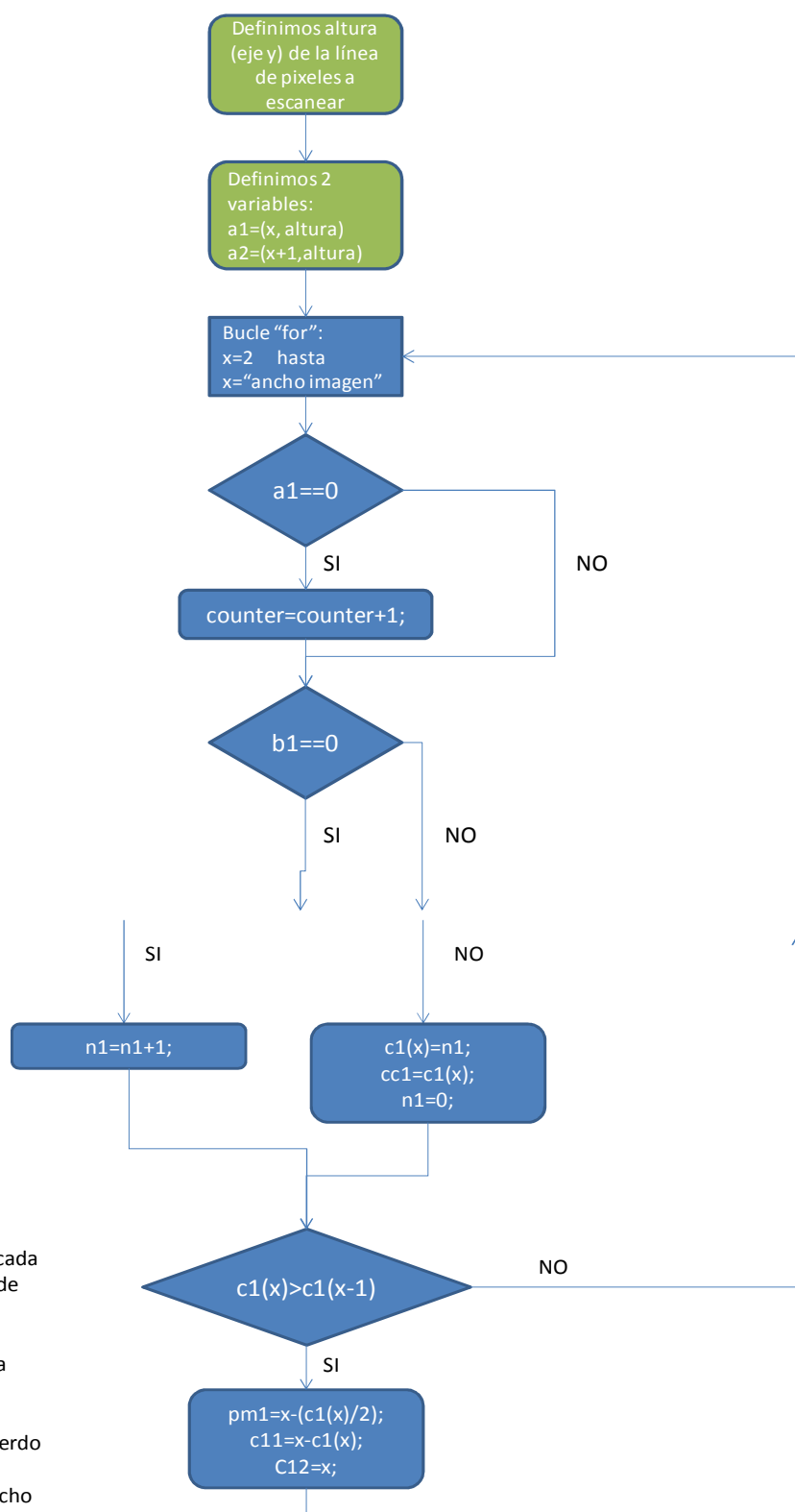


Figura 5.4

Durante el desarrollo de este programa de control, llevé a cabo una serie de correcciones en el cálculo de los puntos que determinan la línea de la trayectoria que tiene que seguir la moto.

El cálculo de la trayectoria mediante la unión de los puntos medios, funciona siempre que la línea negra se mantenga relativamente recta. En caso de que existan curvas pronunciadas, la imagen capturada de la línea negra es casi horizontal y el punto medio de la línea de escáner, no es el punto por el que debería pasar la moto para girar. La moto tiende a salirse por el exterior de la curva.

Este caso se puede apreciar en esta imagen, donde las líneas rojas son las líneas de escáner, los puntos verdes determinan el inicio y fin de cada zona de píxeles negros encontrada por la línea de escaneo y los puntos azules son los puntos medios de los segmentos y cuya unión determina la trayectoria que seguiría la moto.

Se ve que al llegar a una curva pronunciada, la trayectoria tiende a salirse de la línea negra hacia el exterior.

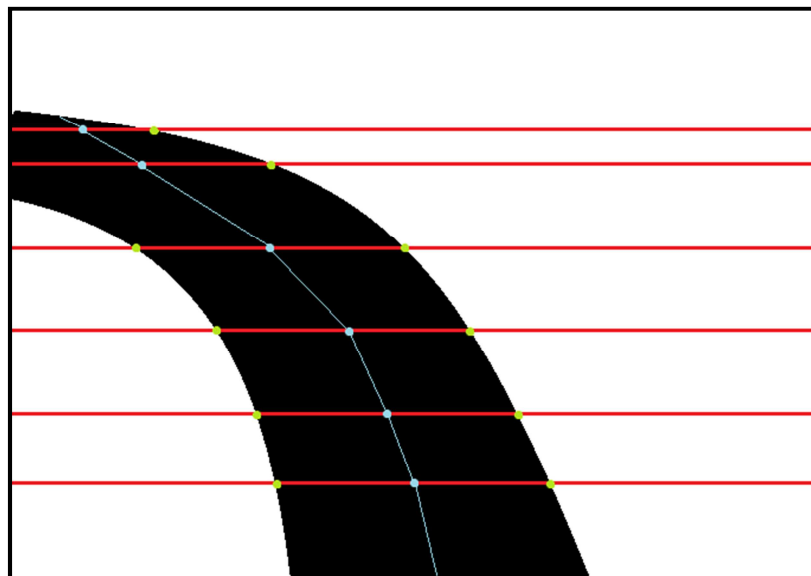


Figura 5.5

Para corregir este problema, en vez de utilizar el punto medio del segmento para el cálculo de la trayectoria, consideré que lo que necesitaba era hallar el centro de masas de la línea negra. Para ello he diseñado un sistema de escáner mediante trapecios.

Aprovechando el programa anterior, que trazaba líneas horizontales y leía los píxeles negros que había en ella para reconocer la línea negra, identificamos los trapecios que forman la unión de los extremos de los segmentos de dos líneas horizontales trazadas consecutivas sobre la línea negra.

De esta forma nos queda la línea negra dividida en una serie de trapecios, como podemos observar en las figuras siguientes.

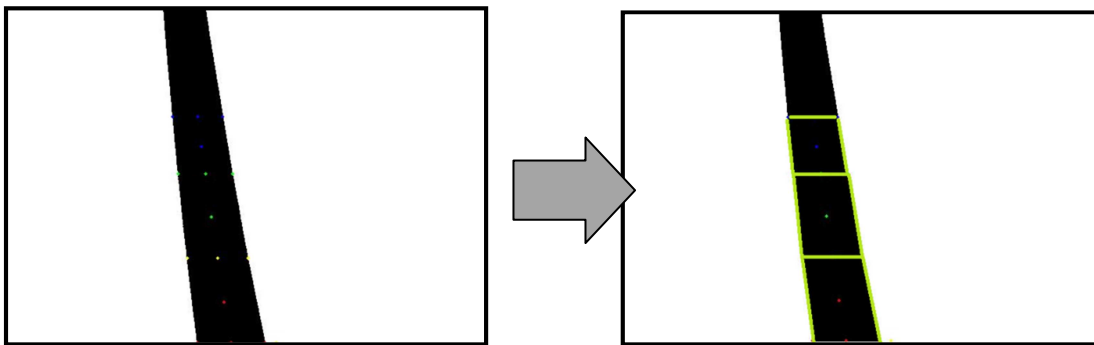


Figura 5.6

A continuación se explica de forma detallada el cálculo del centro de masas de uno de los trapecios.

Mediante el análisis de las líneas horizontales, lo que hacemos es identificar los vértices del trapecio; cada segmento de una línea horizontal identifica dos vértices. Una vez hallados estos puntos en dos líneas consecutivas, ya podemos trazar un trapecio.

Ahora que tenemos el trapecio, tenemos que identificar su centro de masas. Para ello fijaremos una nomenclatura para los puntos del trapecio, de izquierda a derecha y de abajo a arriba, los vértices serán $c11, c12, c21$ y $c22$,; las distancias de sus lados paralelos (bases), de abajo a arriba, serán $cc1$ y $cc2$ y la distancia entre sus bases quedará como $d1$ quedando así el trapecio:

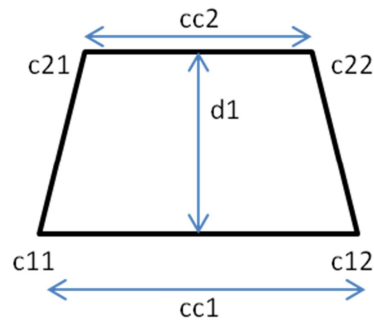


Figura 5.7

Para hallar el centro de masas de forma analítica tendríamos que calcular la masa del trapecio (m), integrando la función que define el trapecio hallaríamos su área y multiplicándola posteriormente por la densidad hallaríamos su masa. También calcularíamos sus momentos M_x y M_y , quedando definido el punto de centro de masas como $C=[M_y/m, M_x/m]$.

En vez de calcularlo de forma analítica, el cálculo lo hago de forma gráfica, que consiste en realizar las operaciones siguientes:

1. Calculo la horizontal de altura h_1 , donde se va a encontrar el centro de masas, desde la base mayor (cc_1), que está definida por la siguiente fórmula:

$$h_1 = d_1 * (cc_1 + 2 * cc_2) / (3 * (cc_1 + cc_2))$$

2. Sumo a la base mayor la distancia de la base pequeña y viceversa; obteniendo los puntos $A = ((c_{11} - cc_2), 0)$ y $B = ((c_{22} + cc_1), d_1)$
3. El centro de masas " cm " del trapecio se encuentra en la intersección de la línea horizontal " $y = h_1$ " con la línea verde que une los A y B descritos anteriormente, quedando como se aprecia en la figura.

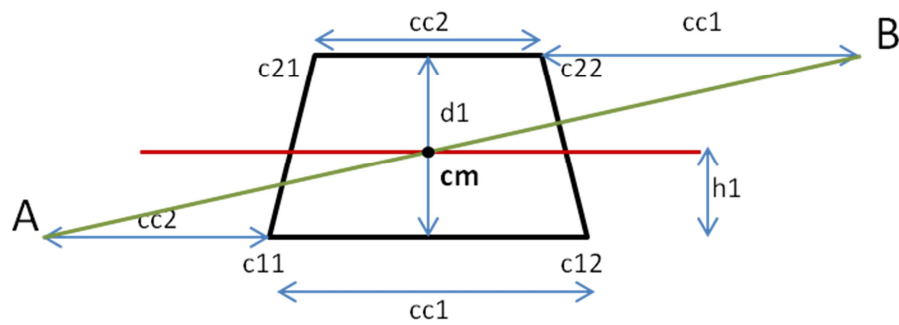


Figura 5.8

La unión de estos centros de masa define la trayectoria de la línea negra.

Con esta corrección el programa de seguimiento final quedaría como sigue:

1. Se lee la imagen de la línea negra proveniente de la cámara en Matlab, se convierte a escala de grises y posteriormente a binario (0=negro, 1=blanco).
2. Se establecen líneas horizontales de escáner en las que se cuenta y detecta el color de los píxeles que contienen. Con esta información se determina el ancho de la línea negra y se calculan los extremos del segmento
3. Se calculan los centros de masa de los trapecios
4. Conocidas las posiciones de estos centros de masa y las distancias entre ellos, se obtiene la trayectoria de la línea negra.
5. Conocida la altura y ángulo de instalación de la cámara y la velocidad de la moto, se calcula el ángulo de giro del motor de dirección requerido para que la moto gire siguiendo la línea negra.

La velocidad de la moto en este proyecto se ha fijado a la máxima que permite el tiempo del ciclo de control. Es decir, a menor velocidad sí es posible controlar la moto ya que el número de ciclos y por tanto el número de giros de la moto por espacio de tiempo es suficiente para que la moto se pueda guiar. Superando esta velocidad máxima la moto recorre demasiado espacio antes de cada giro, por lo tanto la moto se sale de la trayectoria de la línea negra.

Para calcular el ángulo de giro que mandamos a la CPU de Lego he seguido estos pasos:

1. Como necesitamos una referencia fija de dónde está situada la moto para determinar el ángulo de giro, tomamos el punto medio del ancho de la imagen en el eje de abscisas.

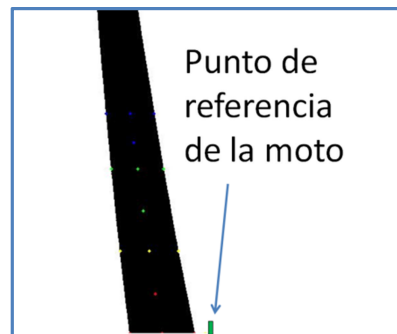


Figura 5.9

2. Sabiendo la distancia que recorre durante cada ciclo de control, sabremos dónde se situará la moto en la siguiente captura, por lo tanto el ángulo de giro es el formado por la línea que une el punto de referencia de la moto y el centro de masas que se encuentra a la distancia que va a recorrer la moto y la horizontal. Quedando como se muestra en la imagen.

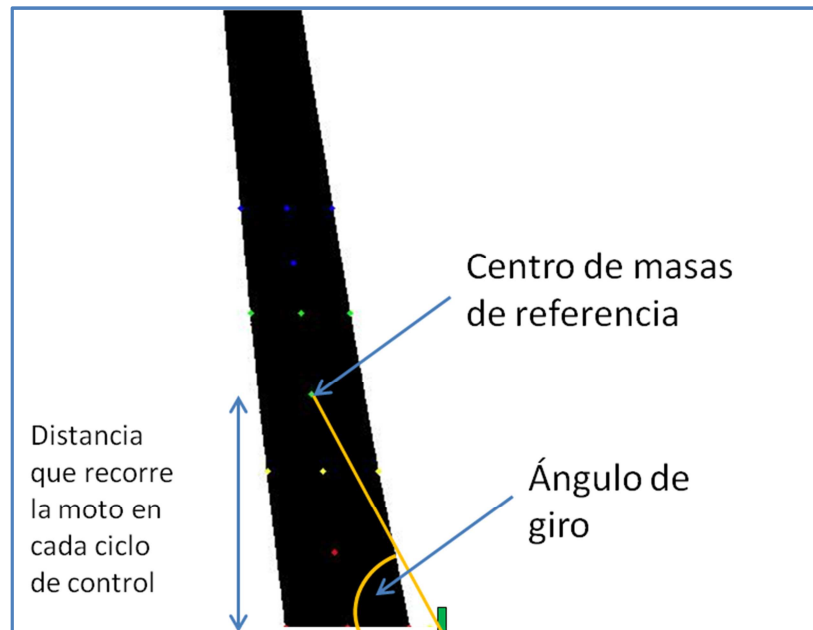


Figura 5.10

Este proceso se ha modelado en el programa de Matlab que he llamado “calc_giro.m” y cuyo código se adjunta en el Anexo 7.1.2.

5.1.3.2. CONTROL DE GIRO

Una vez que tenemos la variable de giro con su valor correspondiente la procesamos siguiendo los siguientes pasos:

1. Si el giro tiene valor positivo, será un giro hacia la derecha y viceversa. Los motores paso a paso que lleva instalada la moto son controlados definiendo su potencia, si esta potencia es negativa o positiva y su ángulo de giro. Con una potencia positiva el motor girará hacia delante y viceversa. En nuestro caso como el motor está tumbado, si le transmitimos una potencia negativa girará hacia la derecha y con una positiva hacia la izquierda. Por lo tanto para un valor de giro hacia la derecha transmitiremos una potencia negativa.
2. Si el giro tiene valor "0", la moto no girará hasta el siguiente ciclo de control si lo precisa.
3. Si el giro presenta un valor no numérico, nos indica que la imagen captada por la cámara no ha sido de la línea negra. Esto podría llegar a ocurrir en giros más

cerrados donde el giro de la moto se produce de forma más lenta que en la captura de la siguiente imagen, ya que la estructura de la moto es bastante endeble y los motores no tienen el suficiente par para afrontar el giro a la velocidad normal. Al ir la moto más lenta o encontrar algún obstáculo, el giro se produce antes, por lo tanto si se capta imagen blanca se procederá a girar la moto $1/4$ del giro realizado anteriormente pero en sentido contrario. He comprobado que en la siguiente imagen ya ha girado lo suficiente como para volver a captar la línea negra y continuar con el seguimiento.

5.2. CALCULO DE LA INCLINACION DE LA MOTO

La parte del control que queremos desarrollar en esta segunda parte del proyecto, es el cálculo de la inclinación que lleva la moto en todo momento, y con ello generar una señal para introducirla en un sistema de control.

Esta variable sería fácilmente calculable si se instalase un giróscopo o un par de sensores infrarrojos uno a cada lado de la moto que nos indicasen la distancia al suelo y así saber cuánto de inclinada está la moto.

Nuestra intención es calcular la inclinación mediante el procesamiento de la imagen que nos manda la cámara.

La primera idea que se me ocurrió, fue tomar como punto fijo en el espacio, el horizonte, y comparar esa línea con una línea horizontal de nuestra imagen; hallando así, mediante el corte de ambas el ángulo de inclinación que lleva la moto. Sería un sistema similar al que llevan los aviones para hallar su inclinación.

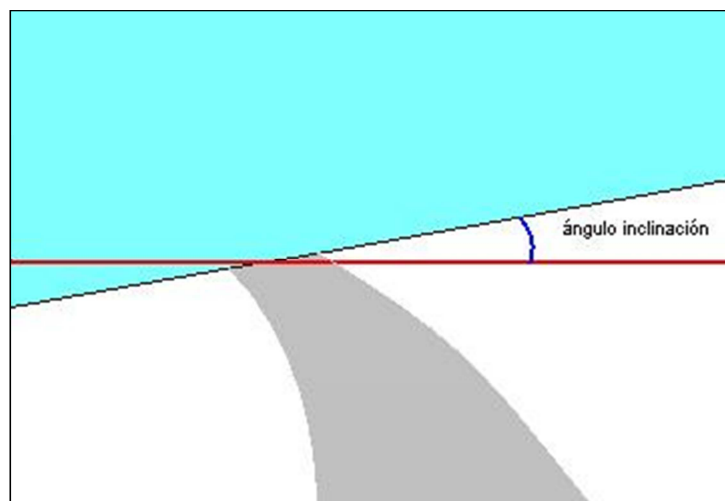


Figura 5.11

El problema que tenemos, por el cual no lo podemos hacer de esta manera, es que la moto es de 15cm de altura y la cámara está colocada de tal forma que lo único que visualizamos es una línea negra sobre el suelo; no está enfocada hacia el horizonte, sino que está inclinada hacia abajo, ya que tenemos que ir tomando imágenes de la línea a una distancia cercana para poder predecir con exactitud por donde debe ir la moto.

Esto nos impide tener un punto de referencia, lo que es necesario para obtener la inclinación que lleva la moto.

Para solventar este problema, dotaremos a la moto de dos punteros láser, los cuales nos servirán de puntos de referencia para hallar la inclinación.

La idea es captar con la cámara la imagen de los punteros laser, identificar los puntos y ver como varían sus posiciones cuando la moto se inclina. Analizaremos las variaciones de posición y a partir de ahí, sacaremos el ángulo de inclinación de la moto.

5.2.1 COLOCACIÓN DE LOS PUNTEROS LASER

Los punteros laser irán colocados por debajo de la cámara, en el mismo plano y enfocando hacia el suelo, con 60º de inclinación sobre la horizontal de la moto; situándose las proyecciones de los punteros en la región que capta la cámara.

En un principio coloqué los punteros paralelos al plano longitudinal de la moto; tras una serie de fotografías captadas con distintos ángulos de giro de la moto, aprecié que esto no podría ser viable, ya que llegaba a un ángulo de inclinación en el que la proyección de uno de los laser se salía de la región que captaba la cámara.

La solución a este problema fue girar 10° sobre el mismo plano cada puntero. El izquierdo se giró 10° hacia la izquierda y el derecho 10° hacia la derecha. Con esta modificación conseguimos que la proyección de los punteros que incide sobre el suelo no salga de la región que capta la cámara. Al estar cada puntero direccionado hacia el lado de inclinación de la moto, reducimos la distancia de impacto del puntero contra el suelo, por lo tanto reducimos así el espacio que abarca la proyección en todo el movimiento de inclinación, pudiendo captar la proyección del puntero en todo momento con la cámara.

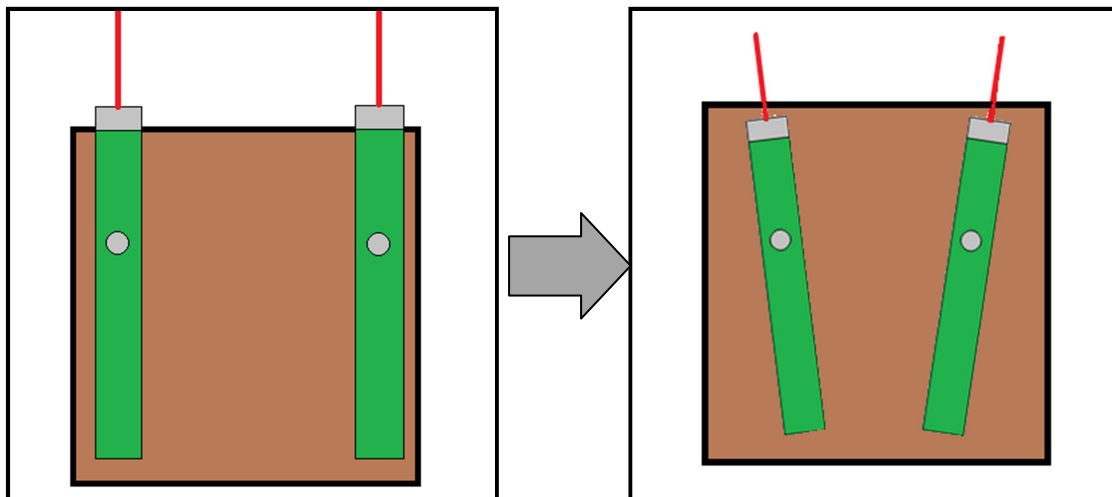


Figura 5.12

5.2.2. RECONOCIMIENTO DE LOS PUNTEROS LASER

Necesitamos determinar mediante el análisis de la imagen captada por la cámara, la situación exacta de los punteros.

La imagen capturada por la cámara cuando tenemos los punteros láser, es similar a esta:



Figura 5.13

Partimos de la imagen en formato RGB (rojo, verde, azul), al ser el láser transmitido por los punteros, de color rojo, lo que vamos a hacer es trabajar solamente con el plano de este color, lo que nos permitirá hallar más fácilmente los puntos de más brillo de este plano, que serán los dos puntos que queremos hallar.

Una vez que tenemos seleccionado el plano rojo, crearemos una matriz binaria que nos separe las zonas más brillantes de las menos brillantes. En principio solo nos tendrían que quedar dos áreas diferenciadas, que serían los dos punteros láser, pero como puede haber zonas de la imagen original que sean de tonalidad roja y que tenga

un brillo que lo pueda llegar a detectar como zona brillante en la matriz binaria, crearemos un filtro para que solo tenga en cuenta las áreas brillantes que superen “x” píxeles, quitándonos así ese ruido que podría aparecer en la imagen binaria. Para identificar estas dos áreas, accedemos a las propiedades de la imagen, sacamos una matriz llamada “área” que contiene la cantidad de píxeles contenidos en cada área identificada, por lo tanto mediante el comando “max(area)” (este comando nos devuelve el valor más alto) identificaremos el área de mayor tamaño, posteriormente hallaremos su centroide; ya tendríamos el punto de uno de los láser. Para hallar la segunda área más grande, que es la que produce el otro puntero láser, sobrescribo el valor máximo de la matriz “area” por ‘0’, lo que hace que si ejecuto de nuevo el comando “max(area)” me devuelva el segundo área de mayor tamaño, hallamos el centroide de este nuevo área y ya tendríamos identificados los dos puntos donde inciden nuestros punteros láser.

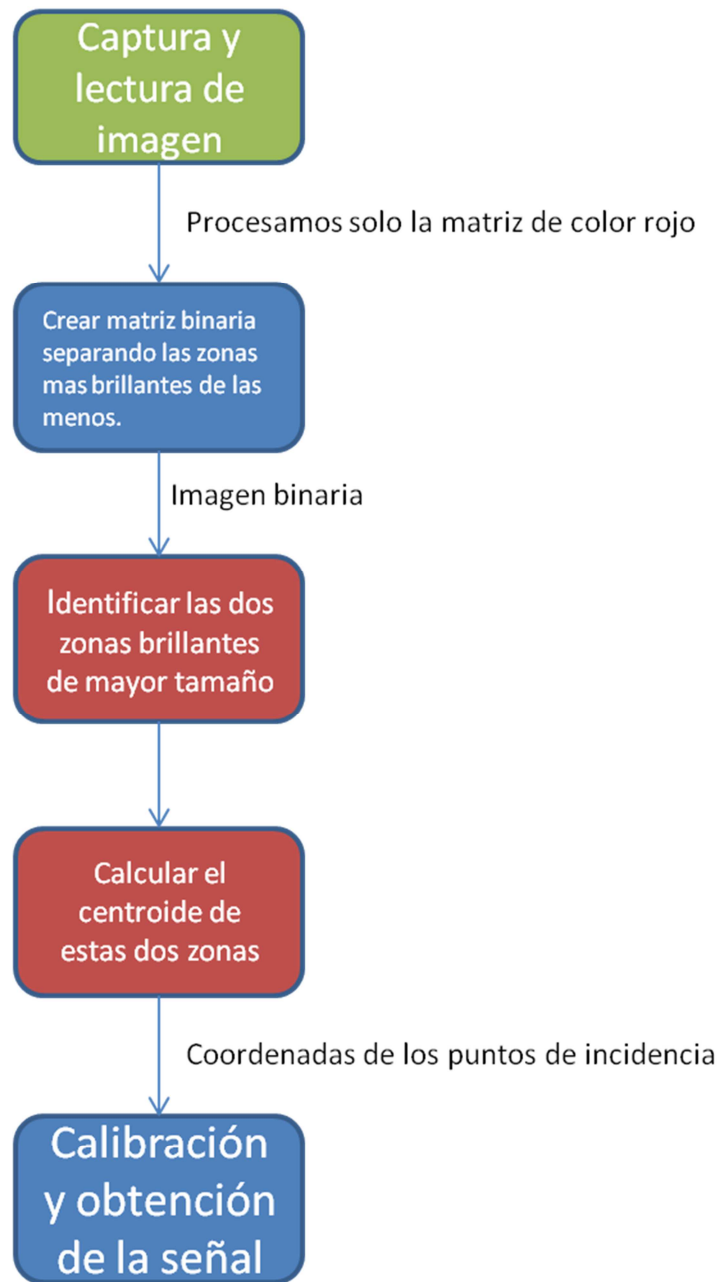


Figura 5.14

Una vez compilado el programa, cuando lo corremos, este nos devuelve la posición en coordenadas cartesianas de los dos puntos. Tomando la altura de cada punto como referencia, vamos a proceder a hacer una calibración. Para ello he tomado una serie de fotografías con la moto sin inclinar y luego inclinándola $+10^\circ$, $+20^\circ$, $+30^\circ$ y $+45^\circ$.

El máximo lo he puesto en 45° ya que una moto de calle no llega a inclinarse más sin apoyar las rodillas en el suelo.

Para la calibración se va a proceder de la siguiente forma:

1. Se captura y procesa la imagen y se devuelven las posiciones de los puntos.
2. Registramos la altura de los puntos, llamaremos punto izquierdo al más situado a la izquierda y punto derecho al otro.
3. Restamos el valor de la altura del punto izquierdo a la del derecho, en la tabla le llamaremos "diferencia".
4. Una "diferencia" de "0" significa que la moto no se encuentra inclinada, una "diferencia" positiva nos indica que la moto se encuentra inclinada hacia la derecha y una "diferencia" negativa hacia la izquierda.

La siguiente imagen corresponde a una inclinación de la moto de 30º grados hacia la izquierda. Podemos apreciar que la altura del punto izquierdo es menor a la del derecho, ya que al estar girando hacia la izquierda el puntero láser izquierdo se encuentra más cercano al suelo, por tanto su proyección incide a menos distancia que la proyección del láser derecho.

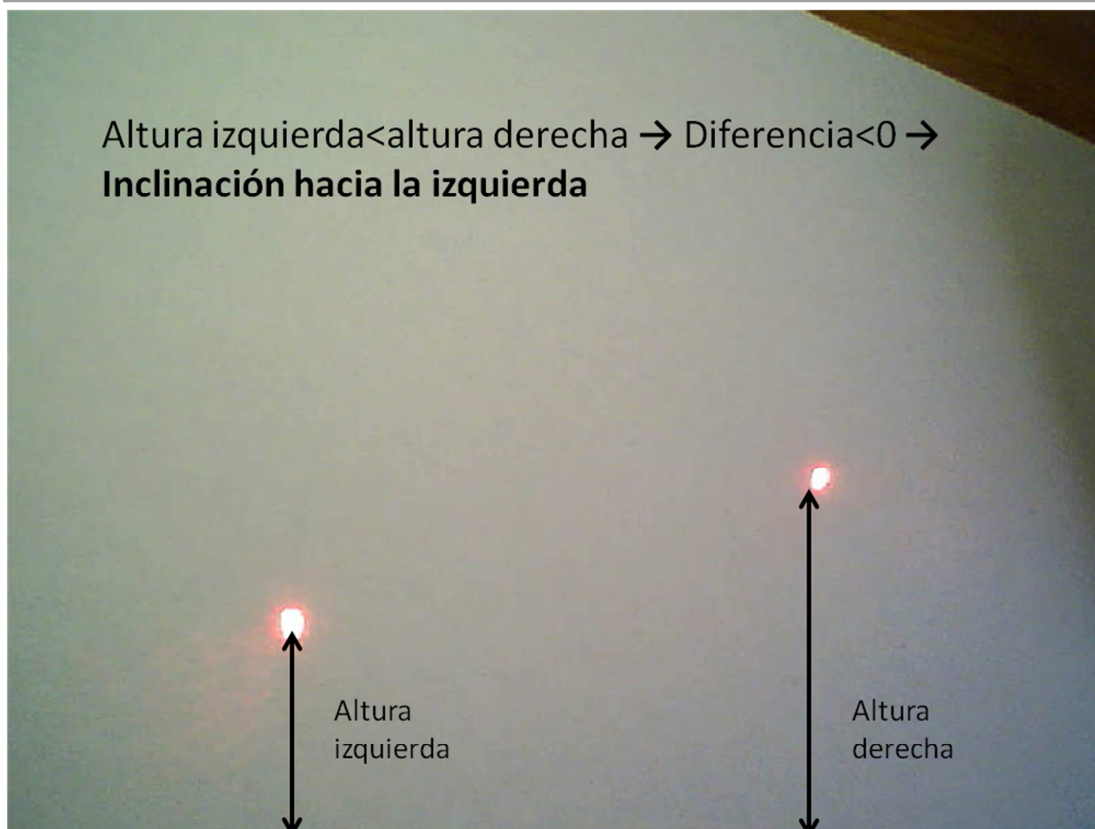


Figura 5.15

En la siguiente imagen veremos el caso contrario, la moto estará inclinada hacia la derecha, en este caso 45° , situándose la proyección del puntero derecho mucho más cercana que la del izquierdo.



Figura 5.16

Por último mostraremos el caso en que la moto no está inclinada y por tanto sus proyecciones se sitúan a la misma altura.



Figura 5.17

He llevado a cabo dos calibraciones, con dos tipos de punteros distintos para verificar, que la variable "diferencia" va tomando valores según la inclinación, que se asemejan a una línea recta. Por lo tanto se puede decir que la función de la variación de la "diferencia" con respecto al ángulo de inclinación de puede asemejar a una recta de pendiente $m \approx 20/50$, para los valores positivos de la inclinación y una recta similar con la misma pendiente pero negativa para los valores negativos de la inclinación.

Sabiendo que la inclinación se asemeja a una recta en los rangos $[-45^\circ - 0]$ y $[0 - 45^\circ]$, se puede utilizar de señal de entrada en un sistema de estabilidad.

Los resultados se ven en las siguiente tablas y gráficas:

Primera calibración							
Dir de inclinación	Ángulo	Altura Izq.	Altura Der	Diferencia		Y Pto izq	Y Pto Der
Recto	0	201	199	2		399	401
Derecha	10	209,31	185,5	23,81		390,69	414,5
Izquierda	10	194,01	200,23	-6,22		405,99	399,77
Derecha	20	169,69	123,21	46,48		430,31	476,79
Izquierda	20	123,95	173,36	-49,41		476,05	426,64
Derecha	30	184,39	105,12	79,27		415,61	494,88
Izquierda	30	114,33	177,79	-63,46		485,67	422,21
Derecha	45	213	81,29	131,71		387	518,71
Izquierda	45	80,91	194	-113,09		519,09	406

Una diferencia positiva (+) nos indicará el giro hacia la derecha de la moto, una diferencia negativa (-) hacia la izquierda.

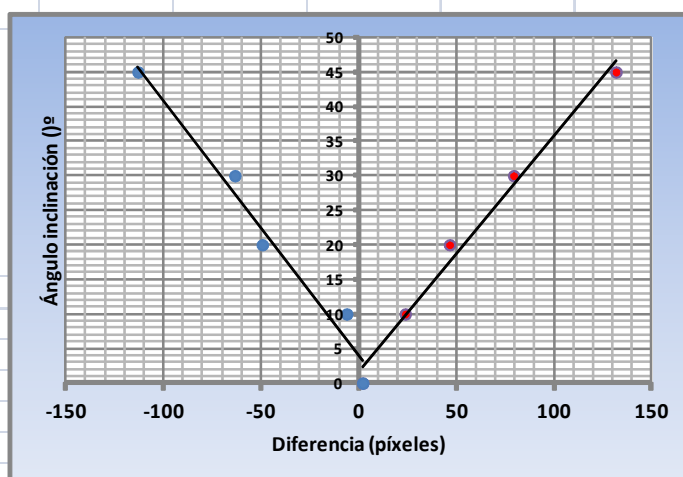


Figura 5.18

Segunda calibración (distintos punteros)							
Dir de inclinac	Ángulo	Altura Izq.	Altura Der	Diferencia		Y Pto izq	Y Pto Der
Recto	0	234,63	223,78	10,85		365,37	376,22
Derecha	10	287,62	255,75	31,87		312,38	344,25
Izquierda	10	259,81	278,2	-18,39		340,19	321,8
Derecha	20	297,37	242,88	54,49		302,63	357,12
Izquierda	20	237,81	296,95	-59,14		362,19	303,05
Derecha	30	269,02	174,2	94,82		330,98	425,8
Izquierda	30	151,07	257,46	-106,39		448,93	342,54
Derecha	45	291,43	139,23	152,2		308,57	460,77
Izquierda	45	87,76	272,27	-184,51		512,24	327,73

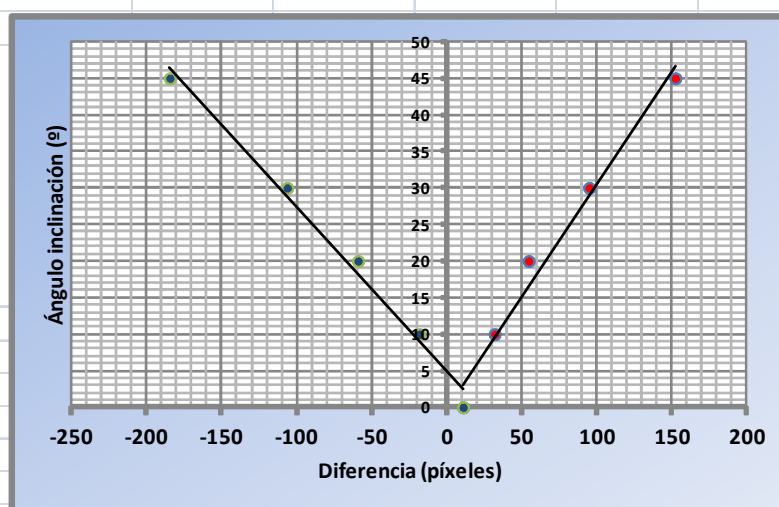


Figura 5.19

6. RESULTADOS Y CONCLUSIONES

Ambos objetivos propuestos para el proyecto se han podido cumplir.

En primer lugar he conseguido implementar en la moto el control de dirección, podemos ver la demostración en el video situado en el anexo xx. Y en segundo lugar he conseguido obtener una señal con el valor de la inclinación la cual podríamos usar posteriormente en un sistema de control de estabilidad.

Aunque los objetivos se han cumplido, se han encontrado limitaciones en la moto y sus componentes que han afectado al desarrollo del proyecto. La moto de Lego está compuesta por piezas de plástico encajadas entre sí mediante uniones de plástico, lo que produce que la moto sea endeble, tenga holguras en la dirección y no se transmita exactamente el giro del motor con la velocidad que se requiere, lo que conlleva arrastrar siempre un error.

Implementando este proyecto sobre una moto de radio control de calidad, se reducirían este tipo de errores. A esto hay que sumarle el tiempo que tarda el envío de la imagen al ordenador mediante el cable USB, el envío de las órdenes mediante Bluetooth a la CPU de Lego y el tiempo que tarda el ordenador en procesar la imagen y obtener el valor de giro. Todo esto produce un tiempo de ciclo bastante alto y que varía centésimas de segundo en cada ciclo, por lo que se ha tenido que fijar en un valor no muy alto el máximo de la velocidad de la moto. Sería una mejora importante si el procesamiento de las imágenes se hiciese en el microcontrolador de la moto, sin tener que pasar por el ordenador, rebajando así el tiempo de ciclo.

Como trabajo futuro se podrían llevar a cabo estas mejoras en una moto de radio control de calidad y complementarlo con un sistema de estabilidad, donde la inclinación se obtenga de la forma que se ha explicado en este proyecto, colocando un par de punteros láser y captando la posición de sus proyecciones con la cámara instalada. El sistema estaría instalado en su totalidad en la moto, procesando los datos y enviando las señales a los actuadores mediante un microprocesador.

7. ANEXOS

7.1. CODIGO

7.1.1. CÓDIGO: control_dir.m

```
vid = videoinput('winvideo', 1, 'YUY2_640x480'); % 'YUY2_160x120'
'YUY2_176x144' 'YUY2_320x240' 'YUY2_352x288' 'YUY2_640x480'
propinfo(vid); %Muestra las propiedades de videoinput
set(vid, 'SelectedSourceName', 'input1'); % selecciona la entrada 1
set(vid, 'FramesPerTrigger', 1); %Numero de frames a tomar cuando se
haga una captura
set(vid, 'TriggerRepeat', Inf); %Numero maximo de frames a tomar
set(vid, 'ReturnedColorSpace', 'rgb'); %Espacio de colores en el que
se trabajara
triggerconfig(vid,'manual'); %Desactiva el modo automatico de toma de
frames

%lego=COM_OpenNXT('bluetooth.ini')
COM_SetDefaultNXT(lego)

M=NXTMotor('AB','Power',-35,'TachoLimit',3000)
M.SpeedRegulation= false
C=NXTMotor('C','Power',35,'TachoLimit',1)
M.SpeedRegulation= false

start(vid);
%encender la camara y alistarse para adquirir imagenes a la memoria
M.SendToNXT();
```

```
i=0;
for i=0:30;
pause(0.1)
frame = getsnapshot(vid);
%distancia('frame')
figure, imshow(frame)
%imshow(frame)

calc_giro
if giro>0;

C=NXTMotor('C','Power',-25)
C.TachoLimit=giro
C.SendToNXT();
elseif giro==0;

elseif isnan(giro)==1
    girorect=round((giro2+giro3)/4)
    if giro<0;
        C=NXTMotor('C','Power',-25)
        C.TachoLimit=girorect
        C.SendToNXT();
    else
        C=NXTMotor('C','Power',25)
        C.TachoLimit=-girorect
        C.SendToNXT();
    end

else
    C=NXTMotor('C','Power',25)
    C.TachoLimit=-giro
    C.SendToNXT();
end

i=i+1;

end
```

```
stop(vid);
```

```
%mostrar en una ventana lo que se esta visualizando por la camara
```

7.1.2. CÓDIGO: calc_giro.m

```
%clear all
%close all
%imagen=imread('paint.jpg');
imagen=frame;
gray=rgb2gray(imagen);
bin=gray>100;
s=size(bin);%tamaño de la imagen
mt=s(2)/2; %punto medio x de la imagen
counter1=0; % contador de pixeles negros, ancho de la linea negra
n1=0;% contador de pixeles negros seguidos
c1=zeros(1,s(2)); %numero de pixels negros seguidos hasta ese punto
counter2=0; % contador de pixeles negros, ancho de la linea negra
n2=0;% contador de pixeles negros seguidos
c2=zeros(1,s(2)); %numero de pixels negros seguidos hasta ese punto
x=1;
pm1=0;%punto medio linea negra escaner 1
pm2=0;%punto medio linea negra escaner 2
counter3=0; % contador de pixeles negros, ancho de la linea negra
n3=0;% contador de pixeles negros seguidos
c3=zeros(1,s(2)); %numero de pixels negros seguidos hasta ese punto
pm3=0;%punto medio linea negra escaner 3
counter4=0; % contador de pixeles negros, ancho de la linea negra
n4=0;% contador de pixeles negros seguidos
c4=zeros(1,s(2)); %numero de pixels negros seguidos hasta ese punto
pm4=0;%punto medio linea negra escaner 4
d1=0;%distancias entre las lineas de escaner
d2=0;
d3=0;

cc1=0;
cc2=0;
cc3=0;
cc4=0;
for x=2:(s(2)-1); %
    a1=bin((s(1)-1),x); %(y,x)no se porque en matlab están cambiadas
    b1=bin((s(1)-1),x+1);
    if a1==0;
        counter1=counter1+1;
        if b1==0;
            n1=n1+1;
        else
            c1(x)=n1;
            cc1=c1(x);
            n1=0;

            if c1(x)>c1(x-1);% solo se pasa el valor del punto
                medio, si el grosor de linea analizadada es mayor que la anterior
```

```
        pm1=x-(c1(x)/2);
        c11=x-c1(x);%guardamos el valor de n1
        c12=x;

    end

end

end

for x=2:(s(2)-1);
    a2=bin((round(s(1)*0.75)),x); %(y,x)no se porque en matlab están
cambiadas
    b2=bin((round(s(1)*0.75)),x+1);
    if a2==0;
        counter2=counter2+1;
        if b2==0;
            n2=n2+1;
        else
            c2(x)=n2;
            cc2=c2(x);%guardamos el valor de n2
            n2=0;

            if c2(x)>c2(x-1);% solo se pasa el valor del punto
medio, si el grosor de linea analizadada es mayor que la anterior

                pm2=x-(c2(x)/2);
                c21=x-c2(x);
                c22=x;

            end

        end
    end

for x=2:(s(2)-1);
    a3=bin((round(s(1)/2)),x); %(y,x)no se porque en matlab están
cambiadas
    b3=bin((round(s(1)/2)),x+1);
    if a3==0;
        counter3=counter3+1;
        if b3==0;
            n3=n3+1;
        else
            c3(x)=n3;
            cc3=c3(x);%guardamos el valor de n3
            n3=0;

            if c3(x)>c3(x-1);% solo se pasa el valor del punto
medio, si el grosor de linea analizadada es mayor que la anterior

                pm3=x-(c3(x)/2);
                c31=x-c3(x);
                c32=x;
```

```
end
end
end

for x=2:(s(2)-1);
    a4=bin((round(s(1)/3)),x); %(y,x)no se porque en matlab están
cambiadas
    b4=bin((round(s(1)/3)),x+1);
    if a4==0;
        counter4=counter4+1;
        if b4==0;
            n4=n4+1;
        else
            c4(x)=n4;
            cc4=c4(x);%guardamos el valor de n4
            n4=0;

            if c4(x)>c4(x-1);% solo se pasa el valor del punto
medio, si el grosor de linea analizadada es mayor que la anterior

                pm4=x-(c4(x)/2);
                c41=x-c4(x);
                c42=x;
                imshow(bin);
                hold on
                plot(pm4,(s(1)/3),'b.','MarkerSize',10)
                plot(pm3,(s(1)/2),'g.','MarkerSize',10)
                plot(pm2,(s(1)*0.75),'y.','MarkerSize',10)
                plot(pm1,(s(1)-1),'r.','MarkerSize',10)
                plot((round(s(2)/2)),(s(1)-
1),'y.','MarkerSize',10)

                plot(c42,(s(1)/3),'b.','MarkerSize',10)
                plot(c41,(s(1)/3),'b.','MarkerSize',10)
                plot(c32,(s(1)/2),'g.','MarkerSize',10)
                plot(c31,(s(1)/2),'g.','MarkerSize',10)
                plot(c22,(s(1)*0.75),'y.','MarkerSize',10)
                plot(c21,(s(1)*0.75),'y.','MarkerSize',10)
                plot(c12,(s(1)-1),'r.','MarkerSize',10)
                plot(c11,(s(1)-1),'r.','MarkerSize',10)

                %c11,c12,c21,c22,c31,c32,c41,c42 son los vértices
de
                %los trapecios

            end
        end
    end
end
```

```
d1=round(s(1)-1)-(round(s(1)*0.75)); %distancia entre los puntos pm1 y
pm2
d2=round (s(1)*0.25); %distancia entre los puntos pm2 y pm3
d3=round((s(1)/2))-round((s(1)/3)); %distancia entre los puntos pm3 y
pm4
```

```
syms recta1 recta2 recta3 i j,
```

```
h1=d1*(cc1+2*cc2)/(3*(cc1+cc2)); %Altura del centro de masas
```

```
recta1=solve('((i-(c11-cc2))/((c22+cc1)-(c11-cc2))-((j-1)/(d1-1)))=0',j); %recta de unión de los puntos del 1er trapecio
cm1=solve('((c22 + cc1 - i - c11*d1 + cc2*d1 + d1*i)/(c22 - c11 + cc1 + cc2))-h1=0',i);
cm1=-(c22 + cc1 - c11*d1 + cc2*d1 + c11*h1 - c22*h1 - cc1*h1 - cc2*h1)/(d1 - 1);
```

```
g1=[cm1,h1] %centro de masas
```

```
h2=d1+d2*(cc2+2*cc3)/(3*(cc2+cc3)); %Altura del centro de masas
```

```
recta2=solve('((i-(c21-cc3))/((c32+cc2)-(c21-cc3))-((j-d1)/(d2)))=0',j); %recta de unión de los puntos del 2º trapecio
cm2=solve('(d2*(d1/d2 + (cc3 - c21 + i)/(c32 - c21 + cc2 + cc3)))-h2=0',i);
cm2=((h2 - d2*(d1/d2 - (c21 - cc3)/(c32 - c21 + cc2 + cc3)))*(c32 - c21 + cc2 + cc3))/d2;
g2=[cm2,h2] %centro de masas
```

```
h3=d1+d2+d3*(cc3+2*cc4)/(3*(cc3+cc4)); %Altura del centro de masas
```

```
recta3=solve('((i-(c31-cc4))/((c42+cc3)-(c31-cc4))-((j-(d2+d1))/(d3)))=0',j); %recta de unión de los puntos del 3er trapecio
cm3=solve('d3*((d1 + d2)/d3 + (cc4 - c31 + i)/(c42 - c31 + cc3 + cc4))-h3=0',i);
cm3=((h3 - d3*((d1 + d2)/d3 - (c31 - cc4)/(c42 - c31 + cc3 + cc4)))*(c42 - c31 + cc3 + cc4))/d3;
g3=[cm3,h3] %centro de masas
```

```
plot( cm1,(s(1)-h1),'r.','MarkerSize',10)
plot( cm2,(s(1)-h2),'g.','MarkerSize',10)
plot( cm3,(s(1)-h3),'b.','MarkerSize',10)
```

```
hold off
```

```
%distancias en el eje y entre los centros de masas
ddl=h1
dd2=h2-h1
dd3=h3-h2

pm1
cm1
cm2
cm3
ang1=atan((cm1-pm1)/ddl)*360/(2*pi) %un ángulo negativo nos indica un
giro hacia la izquierda, uno positivo hacia la derecha
ang2=atan((cm2-cm1)/dd2)*360/(2*pi)
ang3=atan((cm3-cm2)/dd3)*360/(2*pi)
angbase=atan((cm2-(s(2)/2))/(h2))*360/(2*pi)

giro=round(angbase)
```

7.1.3. CÓDIGO: calc_inclinacion.m

```
clear all;
close all;
foto=imread('laser5.jpg');
rojo = foto(:, :, 1);%sacamos el plano rojo de la imagen
imshow(rojo)
% creamos una matriz binaria que separe areas brillantes de las
oscuras.
laser = (rojo >= (0.90 * double(max(rojo(:)))));
props = regionprops(bwlabel(laser), 'Area', 'Centroid');
area = [props.Area];
[num_pixels, index] = max(area); %Hallamos el área más grande
index1=index;
area(index)=0;%Borramos el valor del área más grande, correspondiente
con uno de los punteros
[num_pixels2, index] = max(area); %Hallamos el valor del segundo área
más grande, correspondiente con uno de los punteros
index2=index;
imshow(laser)
% Creamos un filtro que nos deje seleccionar de cuantos pixeles es
almenos el area del laser, para que si es muy pequeña lo ignore y no
nos arroje nada, y hallamos el centro de esa area.
threshold =100;
if (num_pixels > threshold) %hallamos los punto x,y corrspondientes al
centroide de cada área
    row = props(index1).Centroid(2)
    col = props(index1).Centroid(1)
else
    row = NaN;
    col = NaN;
end
if (num_pixels2 > threshold)
    row2 = props(index2).Centroid(2)
    col2 = props(index2).Centroid(1)
else
    row2 = NaN;
    col2 = NaN;
end
```

```
pt1=[row,col]
pt2=[row2,col2]
hold on
plot(col,row,'b.','MarkerSize',10)%marcamos los puntos sobre la
imagen
plot(col2,row2,'b.','MarkerSize',10)
hold off
alfa=atan((row2-row)/(col2-col))*360/(2*pi) %hallamos el angulo que
forma el vector de los dos puntos con la horizontal
```


7.2. SECUENCIAS DE IMÁGENES CAPTURADAS EN EL PROCESO

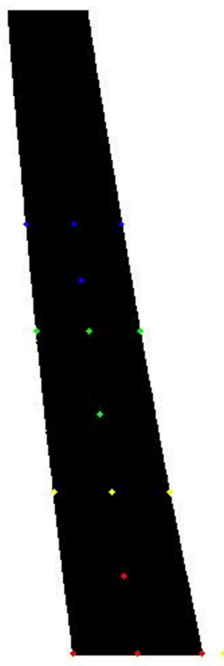


Figura 7.1

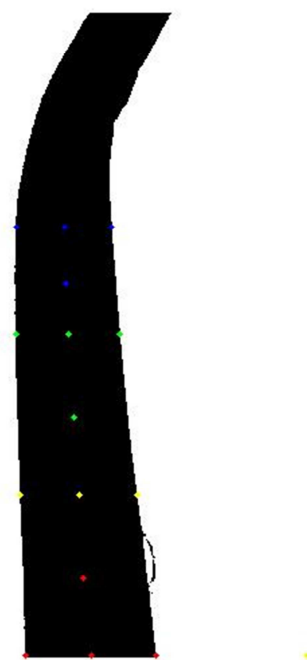


Figura 7.2

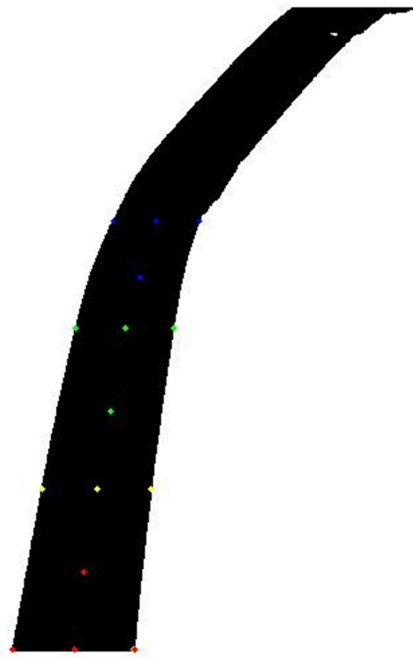


Figura 7.3

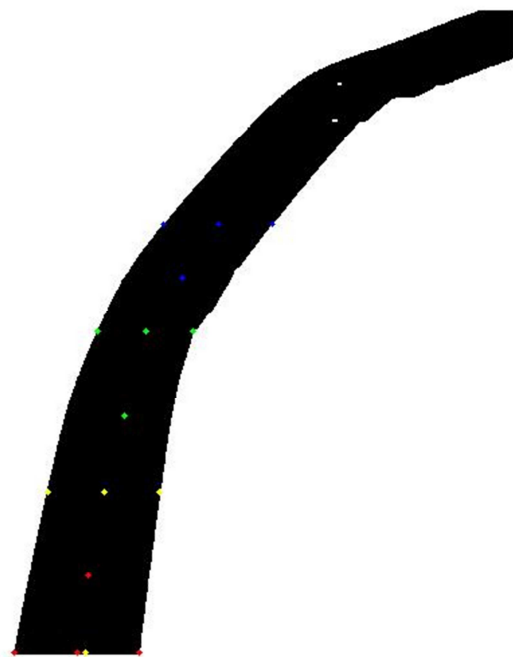


Figura 7.4

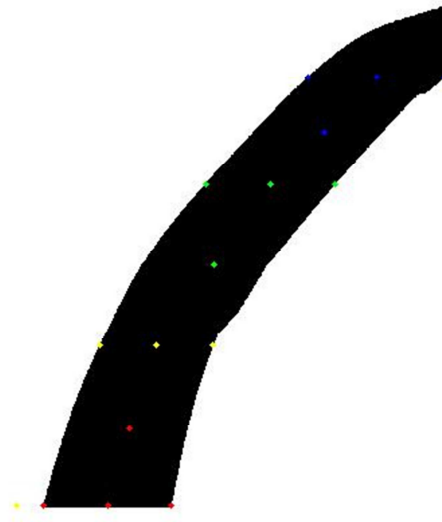


Figura 7.5

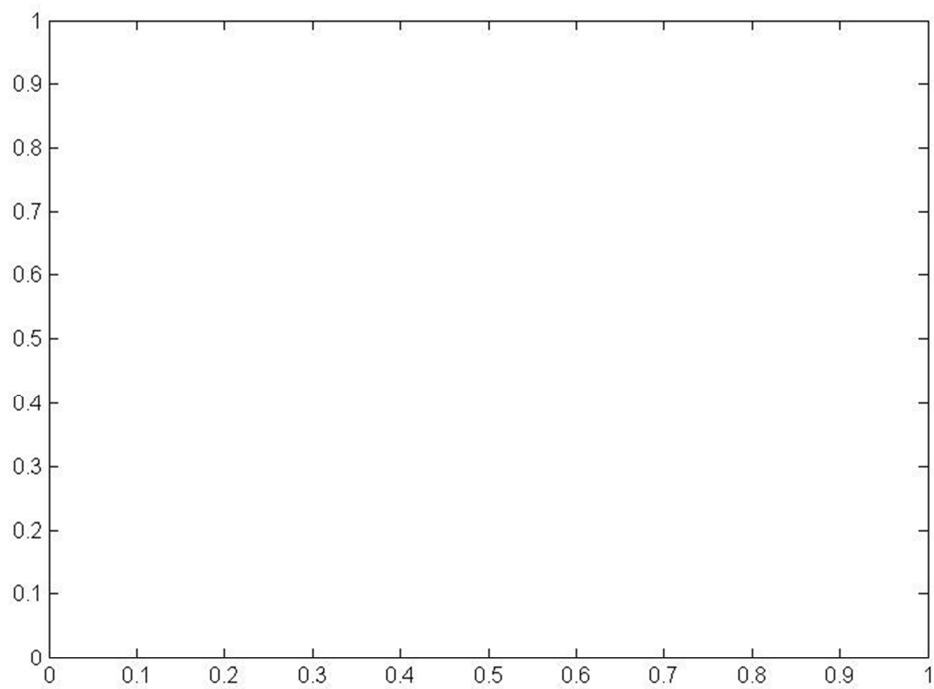


Figura 7.6

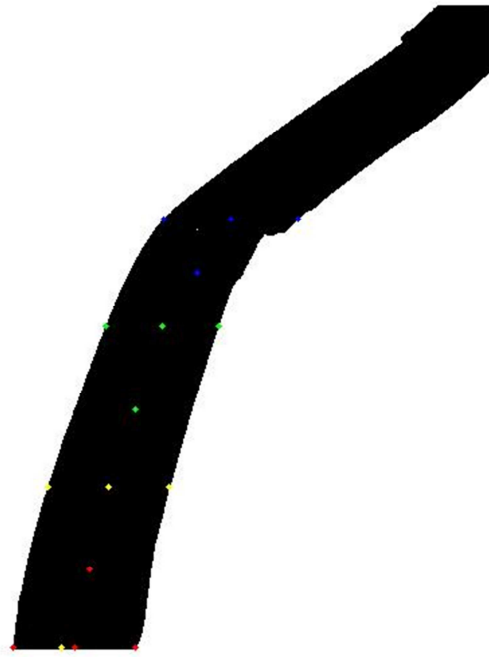


Figura 7.7

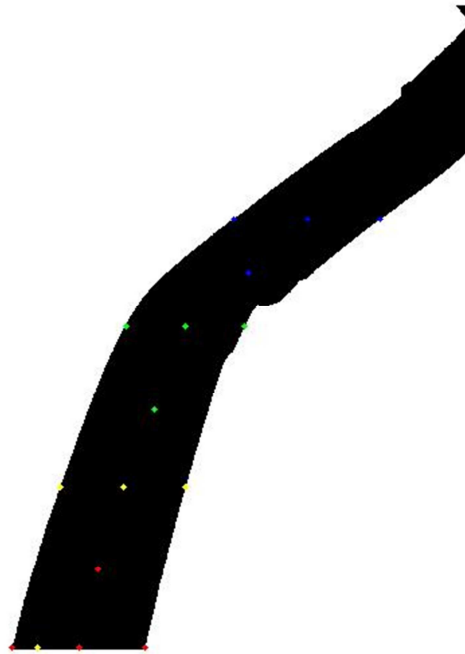


Figura 7.8

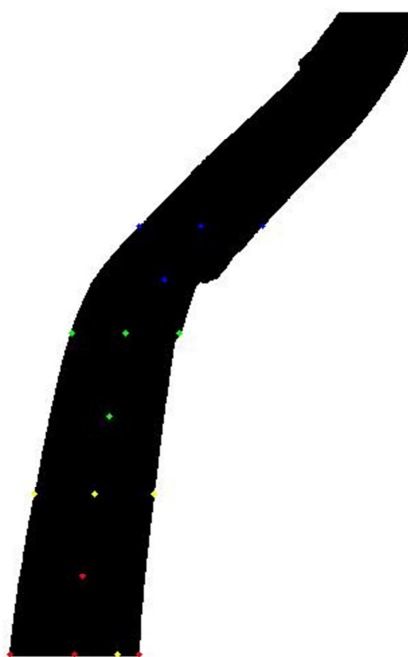


Figura 7.9

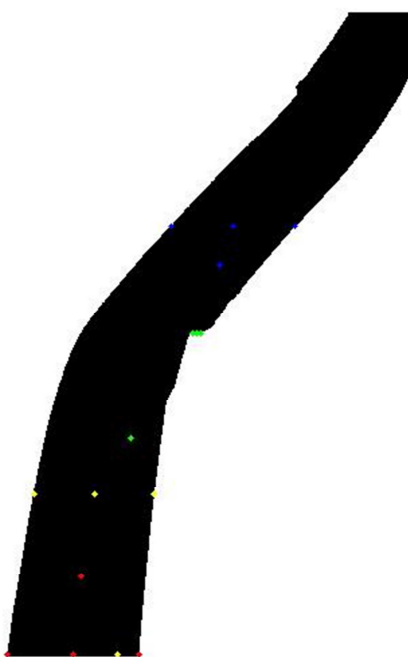


Figura 7.10

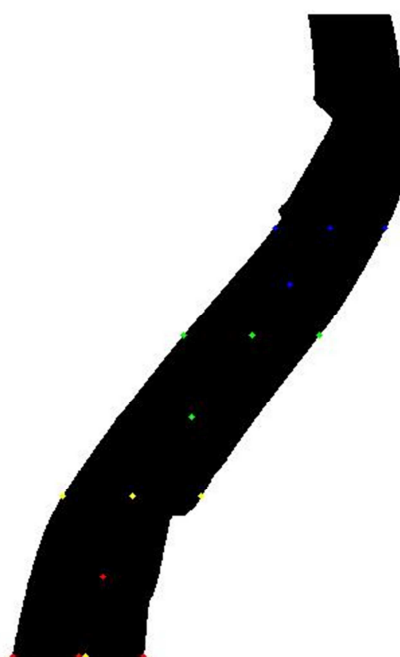


Figura 7.11

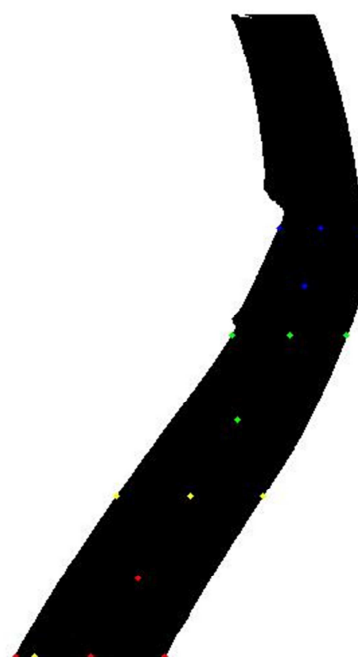


Figura 7.12

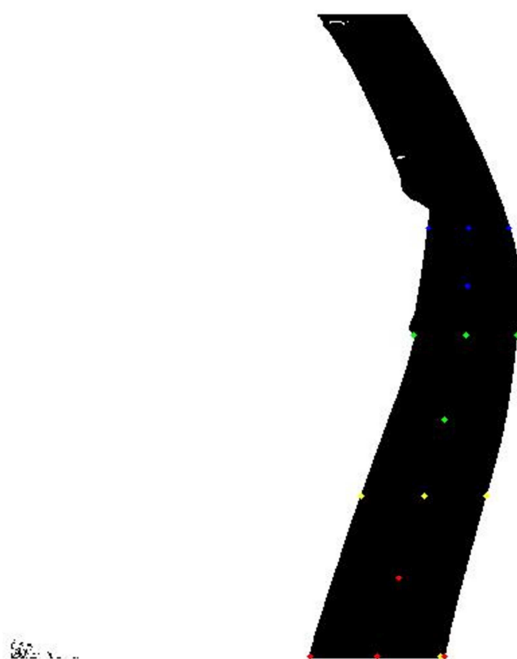


Figura 7.13

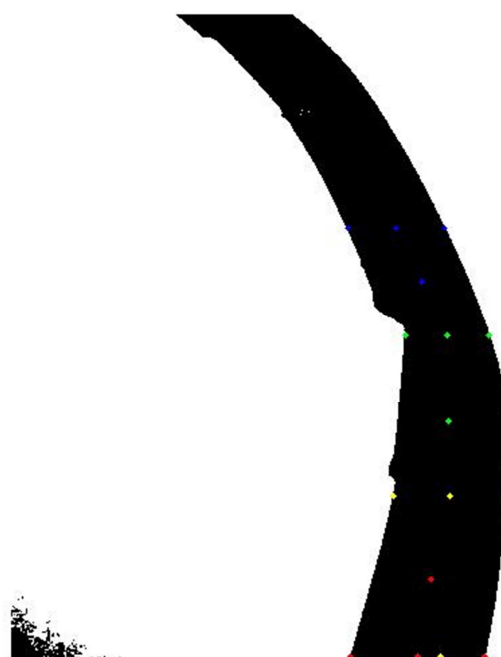


Figura 7.14

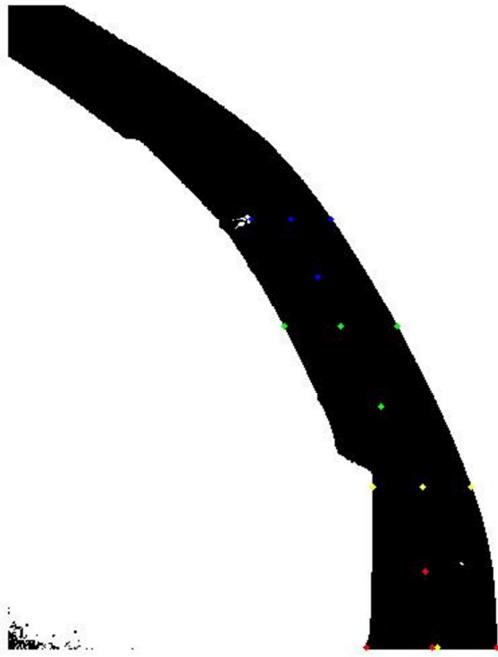


Figura 7.15

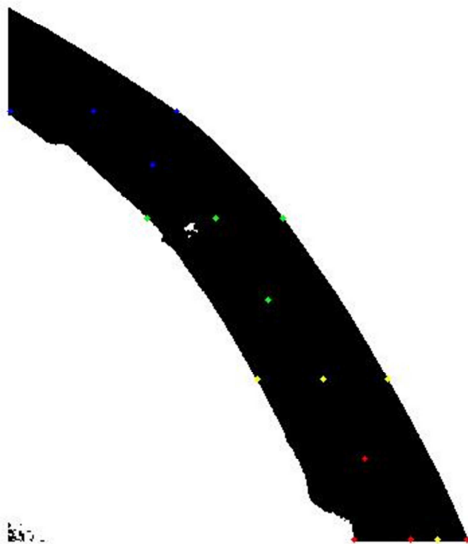


Figura 7.16

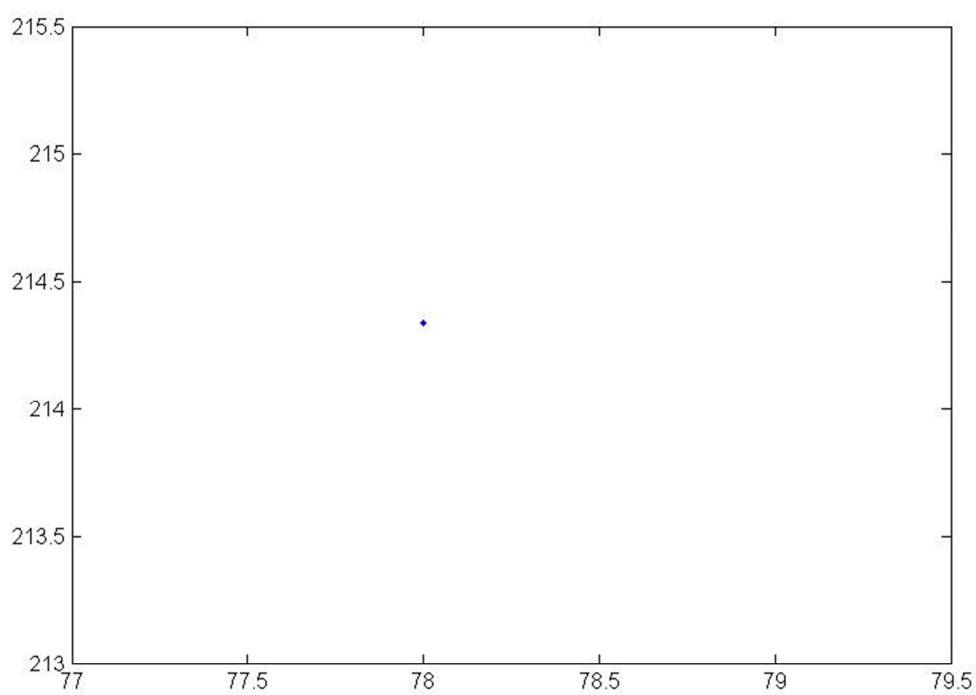


Figura 7.17

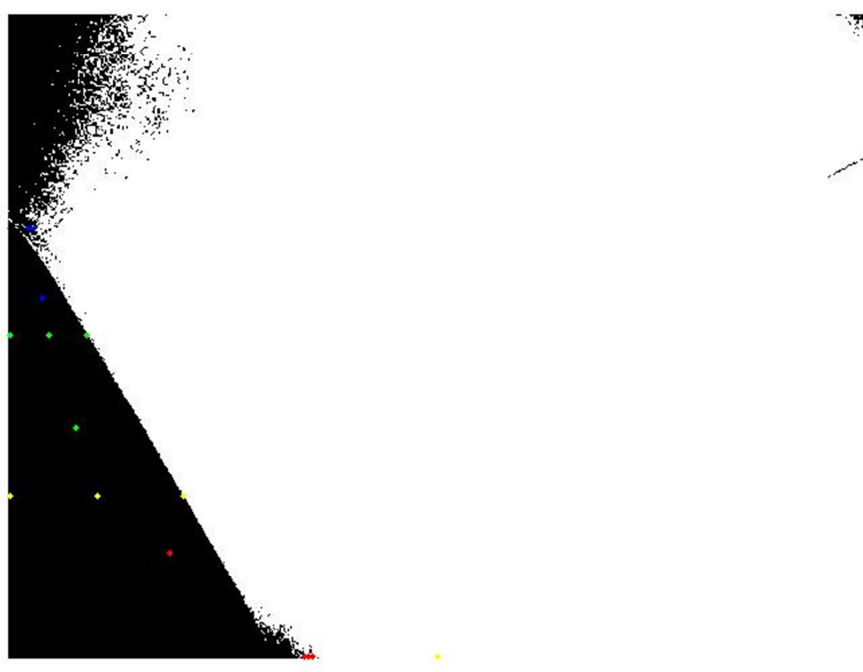


Figura 7.18



Figura 7.19

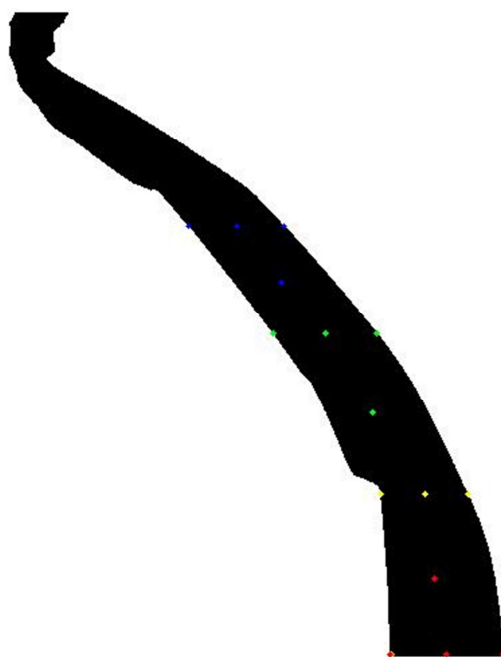


Figura 7.20



Figura 7.21

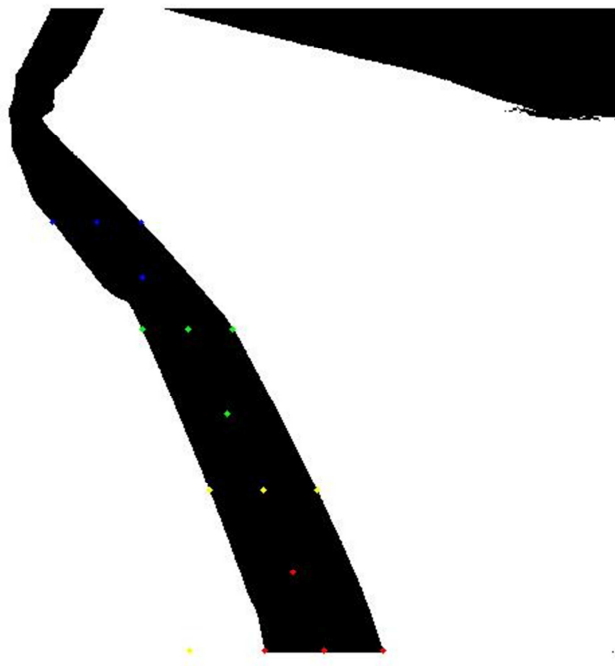


Figura 7.22

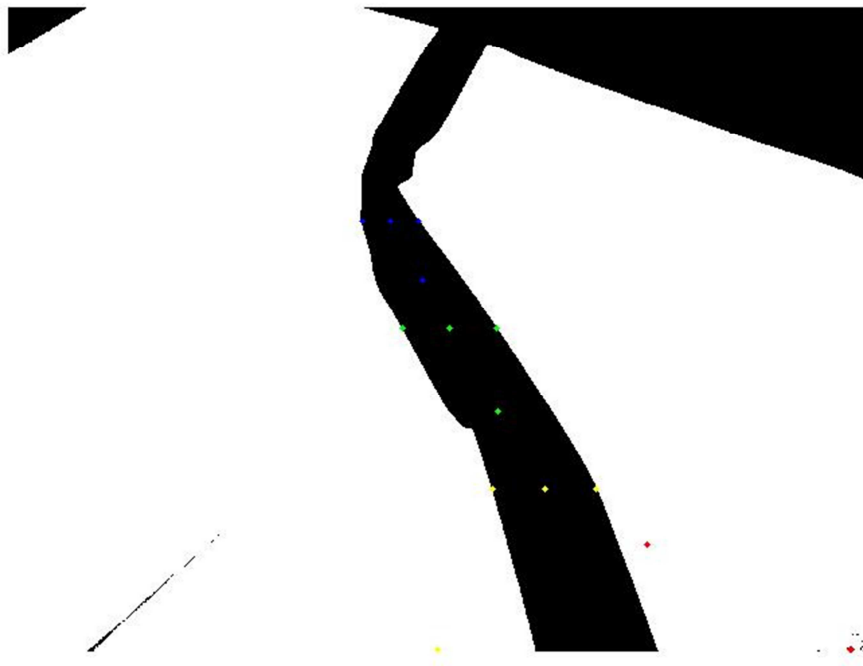


Figura 7.23

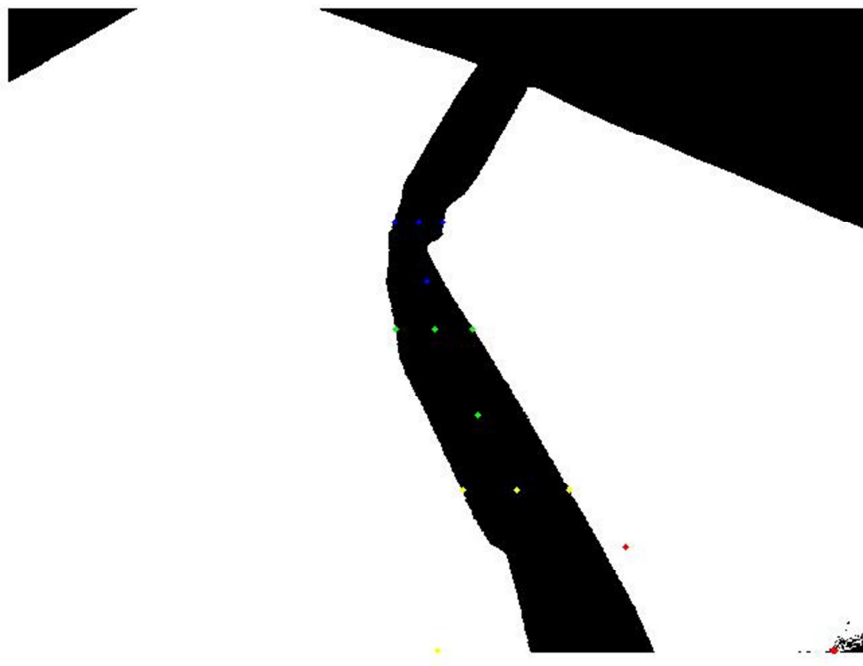


Figura 7.24

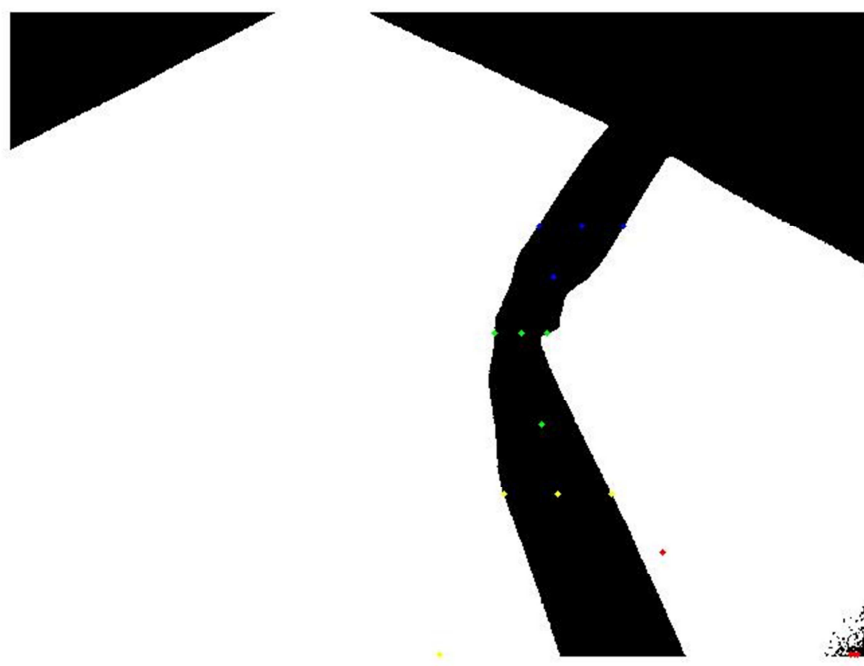


Figura 7.25

7.3. VIDEO DEL SEGUIMIENTO DE LA LÍNEA NEGRA POR LA MOTO

8. BIBLIOGRAFIA

"Tratamiento Digital de Imágenes" - Rafael C. González, Richard E. Woods

<http://www.mindstorms.rwth-aachen.de/> - Página web de la Universidad de Aachen dedicada al Lego Mindstorm.

<http://www.mathworks.es/help/techdoc/ref/f16-6011.html> - Página web de Matlab que incluye el listado de funciones y sus usos.